

Na formação do engenheiro eletricista é fundamental contar com apoio de ferramentas de simulação que facilitem o estudo. Porém na ausência de programas gratuitos com interface gráfica utilizam-se alternativas que requerem dedicar muitas horas de estudo para realizar tarefas.

Neste trabalho implementa-se o programa *DiagramaBarra*, uma interface gráfica simplificada para simulação de sistemas elétricos. Baseado na filosofia de *software* livre, está orientado para facilitar o ensino e a aprendizagem.

Orientador: Fernando Buzzulini Prioste

JOINVILLE, 2019

ANO
2019

GONZALO ANDRÉS ROJAS ÁLVAREZ | DIAGRAMABARRA: UMA INTERFACE PARA
ENSINO E ESTUDO DE SISTEMAS ELÉTRICOS



UDESC

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
CURSO DE MESTRADO PROFISSIONAL EM ENGENHARIA ELÉTRICA

DISSERTAÇÃO DE MESTRADO

**DIAGRAMABARRA:
UMA INTERFACE PARA ENSINO E
ESTUDO DE SISTEMAS ELÉTRICOS**

GONZALO ANDRÉS ROJAS ÁLVAREZ

JOINVILLE, 2019

GONZALO ANDRÉS ROJAS ÁLVAREZ

DiagramaBarra :

Uma interface gráfica para ensino e estudo de sistemas elétricos

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina para obtenção do grau de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Fernando Buzzulini Prioste

JOINVILLE, 2019

A994d Alvarez, Gonzalo Andres Rojas
DiagramaBarra : uma interface gráfica para ensino e estudo de sistemas elétricos / Gonzalo Andres Rojas Alvarez. – 2019.
107 p. : il. ; 30 cm

Orientador: Fernando Buzzulini Prioste

Bibliografia: p. 94-98

Dissertação (Mestrado) -Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Mestrado Acadêmico em Engenharia Elétrica, Joinville, 2019.

1. Engenharia elétrica. 2. Software livre. 3. Desenho eletrotécnico - Métodos gráficos. 4. Sistemas de energia elétrica - Simulação por computador. I. Prioste, Fernando Buzzulini. II. Universidade do Estado de Santa Catarina, Mestrado em Engenharia Elétrica. IV. Título

CDD: 621.3

**DiagramaBarra: Uma Interface Gráfica para Ensino e Estudo de Sistemas
Elétricos**

por

Gonzalo Andres Rojas Alvarez

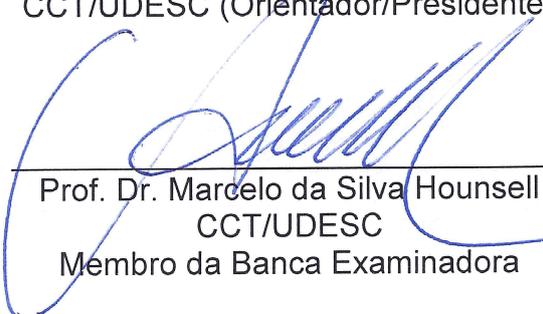
Esta dissertação foi julgada adequada para obtenção do título de

MESTRE EM ENGENHARIA ELÉTRICA

Área de concentração em “Automação de Sistemas”
e aprovada em sua forma final pelo

**CURSO DE MESTRADO PROFISSIONAL EM ENGENHARIA ELÉTRICA
DO CENTRO DE CIÊNCIAS TECNOLÓGICAS DA
UNIVERSIDADE DO ESTADO DE SANTA CATARINA.**


Prof. Dr. Fernando Buzzulini Prioste
CCT/UDESC (Orientador/Presidente)


Prof. Dr. Marcelo da Silva Hounsell
CCT/UDESC
Membro da Banca Examinadora


Prof. Dr. Aleksander Sade Paterno
CCT/UDESC
Membro da Banca Examinadora


Prof. Dr. Djonny Weinzierl
FENÔMENOS APRENDIZAGEM
Membro da Banca Examinadora

Joinville, SC, 28 de fevereiro de 2019.

Dedicado à humanidade e
seus sonhos.

AGRADECIMENTOS

Agradeço a minha família que fez possível estar neste país, me deu apoio, motivação e incentivo

Agradeço o apoio financeiro da UDESC através da bolsa PROMOP.

Agradeço aos professores que me transmitiram novos conhecimentos.

RESUMO

Esta dissertação apresenta um trabalho de desenvolvimento de uma interface gráfica de simulação de sistemas de energia orientada para ensino e aprendizagem. Realiza-se uma investigação da disponibilidade de software de simulação de redes de energia elétrica e tecnologias aplicadas na investigação e produção. Pesquisam-se as atuais vertentes do conhecimento e dos novos paradigmas derivados das mudanças tecnológicas da sociedade com a finalidade de apresentar propostas de atividades pedagógicas com temas transversais com miras na formação integral do profissional. Apresenta-se os passos para o desenvolvimento da interface DiagramaBarra para ser uma ferramenta gráfica de apoio para o professor e para o estudante. Programado em base a plataformas de software livre, DiagramaBarra disponibiliza recursos de suporte para o ensino, ferramentas matemáticas, interface multi-idiomas e multi-plataforma, podendo executar tanto no sistema operacional Windows como Linux. Com base nesta interface se apresenta uma publicação e a aplicação com uso das ferramentas gráficas incorporadas, onde se expõe como alternativa ao software de simulação comercial de alto custo, significando uma vantagem para a adoção por parte de instituições de ensino e profissionais. O produto de este trabalho ficará aberto para que se sigam incorporando facilidades e características no software.

Palavras-chave: Engenharia elétrica, Software livre, Desenho eletrotécnico – Métodos gráficos, Sistemas de energia elétrica – Simulação por computador.

ABSTRACT

This dissertation presents the development of a graphical interface for power system simulation focused on teach and learning. The investigation search availability of power grid simulation software and the last technologies applied on research and production. The survey also follows actual and new paradigms of knowledge founded on the new technological reality of society to propose pedagogical activities with cross objectives that point to an integral formation of professional. DiagramaBarra software was developed as a visual tool to aid teacher and student. It was programmed on open source platforms, brings resources to support learning, mathematical tools, multiplatform interface and custom internationalizations, DiagramaBarra can run both on Windows OS and Linux OS. Article based on this software and newly graphical tools are presented, showing the software as an alternative to commercial software, meaning lower costs for university and professionals. This works will remain open for future modifications and improvements.

Keywords: Electric engineering, Open source software, Electrical drafting--Problems, Electric power systems--Computer simulation.

LISTA DE ILUSTRAÇÕES

Figura 1 – Extrato da lista de tipos de licenças de software livre	29
Figura 2 – Estrutura classe <i>Diagrama</i> e armazenamento lógico	49
Figura 3 – Modelo da classe <i>ODSSClass</i>	50
Figura 4 – Classe <i>Connection</i>	55
Figura 5 – Classe <i>SimulationParameters</i>	56
Figura 6 – Painel de parâmetros de simulação	56
Figura 7 – Classe <i>Results</i>	57
Figura 8 – Fluxograma de montagem e simulação.....	58
Figura 9 – Esquema XML	59
Figura 10 – Modelo de execução simulação	61
Figura 11 – Interface principal.....	62
Figura 12 – Barra de ferramentas.....	63
Figura 13 – Barra de componentes	64
Figura 14 – Painel de propriedades.....	65
Figura 15 – Editor de curvas.....	66
Figura 16 – Gráfico de resultados.....	67
Figura 17 – Painel da matriz de admitância.....	68
Figura 18 – Editor de listas	68
Figura 19 – Webpage do DiagramaBarra no SourceForge	69
Figura 20 – Enquete de percepção do usuário.....	70
Figura 21 – Definições OpenDSS.....	72
Figura 22 – Estrutura OpenDSS	72
Figura 23 – Estrutura do programa DiagramaBarra	73
Figura 24 – Resultados no DiagramaBarra	73
Figura 25 – Circuito equivalente de célula fotovoltaica.....	77
Figura 26 – Curva característica I-V de um módulo	79
Figura 27 – Associação série de dois módulos.....	80
Figura 28 – Associação em paralelo de dois módulos.....	80
Figura 29 – Modelo do PVSystem no DiagramaBarra	81
Figura 30 – Exemplo DiagramaBarra painel fotovoltaico.....	82
Figura 31 – Quadrantes de operação de um inversor	83
Figura 32 – Propriedades InvControler.....	84

Figura 33 – Diagrama do sistema teste	85
Figura 34 – Sistema teste no DiagramaBarra	86
Figura 35 – Curva de radiação solar utilizado	86
Figura 36 – Curva de ativação do tipo de controle VOLT-VAR.....	87
Figura 37 – Curvas de carga e voltagem na barra 10	87
Figura 38 – Curvas de variação de potência ativa e reativa da USF	88
Figura 39 – Variações de tensão na barra 10 com e sem USF para o inversor operando como compensador estático.	89
Figura 40 – Multiplicadores de carga utilizados no caso B	89
Figura 41 – Variações de tensão com e sem USF para o inversor operando como FP unitário.....	90
Figura 42 – Sistema de duas barras e uma linha.....	92
Figura 43 – Diagrama do sistema na qual se realizou análise de sensibilidade	97
Figura 44 – Interface gráfica desenvolvida com os resultados da análise de sensibilidade	98

LISTA DE TABELAS

Tabela 1 – Investimento de licenciamento de software na UDESC	30
Tabela 2 – Lista de <i>Softwares</i> do centro CCT da UDESC	31
Tabela 3 – <i>Software</i> de simulação de redes de potência.....	32
Tabela 4 – Resultados de busca bibliográfica.....	34
Tabela 5 – Principais linhas epistemológicas contemporâneas	37
Tabela 6 – Comparação de bibliotecas para manipulação de diagramas.....	46
Tabela 7 – Elementos da biblioteca OpenDSS incorporados ao Diagrama Barra	50
Tabela 8 – Disciplinas com conteúdos afins com a aplicação do Diagrama Barra e suas ementas	74
Tabela 9 – Exemplo de características elétricas de painel fotovoltaico marca SunPower modelo 305	79
Tabela 10 – Diferentes combinações de operação do controle de inversor..	84

LISTA DE ABREVIATURAS E SIGLAS

ANEEL	Agência Nacional de Energia Elétrica
CCL	<i>Creative Commons License</i>
CCT	Centro de Ciências Tecnológicas
COM	<i>Component Object Model</i>
DEE	Departamento de Engenharia Elétrica
DLL	<i>Dynamic link library</i>
EAD	Educação à distância
EPRI	<i>Electric Power Research Institute</i>
FACTS	<i>Flexible Alternating Current Transmission Systems</i>
FSF	<i>Free Software Foundation</i>
GNU	<i>GNU is Not Unix</i>
GPL	<i>General Public License</i>
HIL	<i>Hardware In the Loop</i>
IaaS	<i>Infrastructure as a service</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
NDA	<i>Non-disclosure Agreement</i>
ODM	<i>Open Model for Exchanging Power Simulation Data</i>
PERD	<i>Program on Energy Research and Development</i>
QSTS	<i>Quasi-Static Time Series</i>
SaaS	<i>Software as a service</i>
SETIC	Secretaria de Tecnologia da Informação e Comunicação
SIN	Sistema Interconectado Nacional
SoaC	<i>System-on-a-Chip</i>
TI	Tecnologia da Informação
TCO	<i>Total cost of ownership</i>
USF	Usina Solar Fotovoltaica
XML	<i>Extensible Markup Language</i>

LISTA DE SÍMBOLOS

I_p Corrente de célula fotovoltaica

V_p Voltagem de célula fotovoltaica

R_S Resistência em serie de célula fotovoltaica

R_P Resistência em paralelo de célula fotovoltaica

SUMÁRIO

1	INTRODUÇÃO	21
1.1	MOTIVAÇÕES E FUNDAMENTOS GERAIS	23
1.2	FUNDAMENTOS ESPECÍFICOS	24
1.3	CONTRIBUIÇÕES DECORRENTES DESTE TRABALHO	26
2	SOFTWARE LIVRE E ENSINO NA ENGENHARIA ELÉTRICA	27
2.1	INTRODUÇÃO	27
2.2	CONCEITO DE SOFTWARE LIVRE	27
2.3	SOFTWARES LIVRES NO ENSINO.	30
2.4	SOFTWARES DE SIMULAÇÃO DE REDES DE POTÊNCIA	31
2.5	REVISÃO BIBLIOGRÁFICA	33
2.6	PARADIGMAS E APRENDIZAGEM	36
2.7	METODOLOGIAS PARA O ENSINO PRATICO-REFLEXIVAS.....	40
2.8	ESPECIFICAÇÕES E REQUISITOS DA PLATAFORMA	42
3	DESENVOLVIMENTO DE UMA FERRAMENTA GRÁFICA DE SIMULAÇÃO	45
3.1	INTRODUÇÃO	45
3.2	FERRAMENTAS DE DESENVOLVIMENTO	45
3.3	DESENVOLVIMENTO DO MODELO DE DADOS.....	48
3.3.1	<i>O elemento ODSSClass</i>	49
3.3.2	<i>O elemento Connection</i>	55
3.3.3	<i>O elemento SimulationParameters</i>	55
3.3.4	<i>O elemento Results</i>	56
3.4	DESENVOLVIMENTO DA SEQUÊNCIA DE SIMULAÇÃO	57
3.5	DESENVOLVIMENTO DO MODELO DE ARQUIVOS	58
3.6	DESENVOLVIMENTO DO MODELO DE EXECUÇÃO	59
3.7	INTERFACE	61
3.7.1	<i>Barra de Ferramentas</i>	63
3.7.2	<i>Barra de Componentes</i>	63
3.7.3	<i>Painel de Propriedades</i>	64
3.8	FERRAMENTAS ADICIONAIS DESENVOLVIDAS	65
3.8.1	<i>Editor de comportamento variável</i>	66
3.8.2	<i>Painel de gráficos de resultados</i>	66
3.8.3	<i>Painel da matriz de admitância</i>	67
3.8.4	<i>Editor de grupos</i>	68
3.9	DISTRIBUIÇÃO E DOCUMENTAÇÃO	69

4	USO DO PROGRAMA DIAGRAMABARRA.....	71
4.1	DIFERENÇAS E MELHORIAS	71
4.2	APLICAÇÃO NO ENSINO	74
4.3	EXEMPLO DE APLICAÇÃO	76
4.4	APLICAÇÃO DO DIAGRAMABARRA NO ANALISE DE PERTURBAÇÕES /SENSIBILIDADE	91
5	CONCLUSÕES	99
5.1	CONCLUSÕES E CONSIDERAÇÕES FINAIS	99
5.2	PROPOSTAS PARA TRABALHOS FUTUROS	100
	REFERENCIAS.....	101
	APÊNDICE A.....	106
	APÊNDICE B: ATIVIDADES PEDAGÓGICAS USANDO DIAGRAMABARRA	110
	ATIVIDADE 1: APRENDER CONCEITOS DE LINHA DE TRANSMISSÃO, CARGA E GERAÇÃO	110
	ATIVIDADE 2: IMPORTÂNCIA DE FATOR DE POTENCIA E CORREÇÃO	111
	ATIVIDADE 3: CARGA BALANCEADA E DESBALANCEADA	112
	ATIVIDADE 4: FLUXO DE POTENCIA DE UM SISTEMA DE 3 BARRAS.....	113
	APÊNDICE C: FORMATO ARQUIVOS DE IDIOMAS	114

1 INTRODUÇÃO

A humanidade durante toda a sua história tem realizado avanços no conhecimento, permitindo a difusão de cultura, o progresso científico e um melhor entendimento dos processos humanos. A fabricação de ferramentas e máquinas permitiu o ser humano realizar tarefas além das suas capacidades físicas. No limiar do século XX, o computador tem-se convertido no principal instrumento do avanço tecnológico. O massivo uso dos computadores para todo tipo de tarefas tem dado passo a uma maior difusão e intercâmbio do conhecimento, novas formas de comunicação, novas visões da realidade e a coexistência de múltiplos paradigmas. A distribuição de ferramentas computacionais na forma de *tablets* e *smartphones*, a conectividade brindada pela Internet e a conectividade 3G/4G/5G, têm mudado as formas do relacionamento humano, as formas de comunicação e a forma de como se escreve a história. No âmbito educacional surge a Educação à distância (EAD) e o estudante tem acesso aos recursos de aprendizagem desde qualquer lugar do planeta, existindo somente limitações idiomáticas.

Neste cenário, a estratégia das empresas de *software* tem evoluído conforme com os tempos, mudando seu produto desde o modelo “mercadoria” para um modelo de negócios tipo “serviço”. Os novos conceitos *Software-as-a-Service* (SaaS) e *Infrastructure-as-a-Service* (IaaS) (BUYA, 2010), apresentam-se como alternativas ao cliente para usufruir da ferramenta, onde o cliente já não paga pelo *software* para ser instalado no seu computador, mas pelo serviço de uso do *software* por assinatura.

De forma paralela ao *software* comercial, surge o *software* livre a partir dos ideais da universalidade do conhecimento e da necessidade de modificar programas para cumprir tarefas a medida. O movimento *software* livre disponibiliza aplicativos com funcionalidades similares aos comerciais mediante descarga gratuita. As aplicações que seguem este princípio têm evoluído e melhorado através do trabalho voluntário da comunidade, favorecendo o crescimento da web e a democratização do conhecimento. (OLIVA e MELO, 2009)

As instituições educacionais, na sua função de formadoras de profissionais, devem estar atualizadas na aplicação e uso das tecnologias tanto no âmbito acadêmico como no administrativo e logístico. Isto requer um grande investimento

no caso de considerar *software* comercial, portanto trata-se de fomentar o uso de *software* livre e o desenvolvimento de soluções. (REITORIA, 2018;2017;2016;2015)

O mundo presenciou nas últimas décadas a substituição do envio de cartas pelo mensageiro instantâneo, dos livros pelo *e-book*, da lousa pelo *PowerPoint*, representando uma mudança na forma de entregar o conhecimento na formação do aluno. O atual aprendiz nasceu numa época de coloridos ícones gráficos. Porém o professor é de uma ou duas gerações anteriores. Essas mudanças em tão curto prazo foram notadas no decorrer do século XX percebendo-se que os tradicionais paradigmas da ciência deviam ser revisados continuamente. (MOREIRA e MASSONI, 2011)

Outro aspecto que motiva este trabalho são os índices de evasão revelados pelo estudo de Davok e Bernard (2016), que chegam a atingir 50% nos cursos da engenharia elétrica. O caráter abstrato dos conteúdos conduz a dificuldades na compreensão da matéria mas que podem ser superadas com ferramentas didáticas. Convém dispor de recursos para o professor na tarefa de dar motivação à busca do conhecimento. Perante esta situação, surge o interesse de gerar uma ferramenta educacional que seja mais adequada aos novos tempos de contínua atualização.

Propõe-se como principal objetivo deste trabalho o desenvolvimento de uma interface gráfica de simulação que permita uma melhor compreensão do conteúdo e sirva de ferramenta base para a formação técnica e superior no curso de engenharia elétrica, inicialmente na disciplina de Sistemas Elétricos de Energia. A plataforma proposta segue os princípios da filosofia do *software* livre. Um objetivo considerado de importância foi escolher um ambiente de programação que tenha suporte no tempo e de essa maneira, pretende-se que o trabalho realizado possa ter futuras modificações. Sendo assim, este documento divide-se em cinco capítulos.

Neste capítulo introdutório proporcionam-se os principais objetivos e motivações do presente estudo. Apresentam-se as características dos componentes externos utilizados no desenvolvimento.

No Capítulo 2 apresentam-se os conceitos de *software* livre e na sequência, um estudo do estado atual do uso de tecnologias livres a través de uma revisão bibliográfica sobre softwares livres de simulação de sistemas elétricos de energia. Posteriormente se expõe uma pesquisa dos paradigmas contemporâneos na procura do processo de construção do conhecimento. Revisam-se as estratégias de ensino pratico-reflexivo. A partir de um resumo das epistemologias, busca-se consolidar

critérios para dar forma a atividades educativas suportadas na tecnologia tendo em vista a aplicação de metodologias pedagógicas modernas. Conclui o capítulo com as especificações e requerimentos para desenvolver a plataforma de simulação proposta, de forma a favorecer a aprendizagem e a geração de material de ensino.

No Capítulo 3 é apresentado o trabalho de programação efetuado, a descrição da adaptação das bibliotecas importadas, a estrutura lógica de suporte para a construção do software, as ferramentas de ajuda incorporadas e os desafios encontrados no processo. Conclui-se com a plataforma utilizada para distribuição a ser empregada.

O Capítulo 4 é dedicado à aplicação *software* de simulação realizado. Compara-se com a alternativa OpenDSS¹. Apresentam-se atividades suportadas nas teorias de ensino revisadas no segundo capítulo e uma aplicação realizada para uma publicação. Revisa-se o uso na adaptação de uma nota técnica. Resume-se uma análise geral do avanço no contexto de interesse.

No Capítulo 5 resumem-se as conclusões e resultados de cada parte do trabalho e se propõem possíveis estudos futuros sobre a plataforma implementada.

1.1 MOTIVAÇÕES E FUNDAMENTOS GERAIS

A principal motivação deste trabalho é a situação generalizada de diminuição do investimento na educação por parte do governo e das iniciativas privadas. Em consequência, as instituições educativas tendem a reduzir despesas e gastos operativos. Uma das despesas a ser impactadas no futuro será o licenciamento de software. A segunda motivação é disponibilizar uma ferramenta que possa ser utilizada pelo professor e o aprendiz no processo educativo, assim como pelo profissional no ambiente laboral. Geralmente, a academia dispõe de recursos para os alunos com finalidade de treinamento e capacitação. Uma vez que o aluno se gradua e entra no mercado laboral, perde o acesso a esses recursos e termina recorrendo à pirataria de software comercial. Estas razões fundamentam a eleição do desenvolvimento dentro da filosofia do *software* livre.

Os atuais paradigmas e diferentes teorias de aprendizagem surgem de mudanças no pensamento do ser humano. O acesso às novas tecnologias

¹ OpenDSS <http://www.sourceforge.net/electricdss> . Acesso em 10 de fevereiro 2019

influenciam a cultura e os sonhos do habitante, que procura uma formação que possa entregar um maior domínio de esta ciência. Mas existe uma alta taxa de abandono e desistência nas faculdades de engenharia. Será necessário adaptar a metodologia do professor para motivar o estudante ou se requer uma análise dos conhecimentos entregados na formação do profissional. As dificuldades de aprendizado apresentadas decorrem do tempo disponível para resolver tarefas ou da falta de compreensão dos conteúdos. Podem resolver-se problemas dos professores com recursos pedagógicos tecnológicos sem restrições e personalizáveis. Para conduzir estas dúvidas a soluções concretas, apresenta-se uma revisão de paradigmas atuais e propõem-se estratégias e atividades pedagógicas visando o uso da interface DiagramaBarra como recurso educativo.

1.2 FUNDAMENTOS ESPECÍFICOS

Neste documento, referira-se ao uso de OpenDSS como ferramenta de cálculo de sistemas de energia. A eleição de tal ferramenta se baseia principalmente no código livre disponível, na variedade de aplicações nas quais pode ser utilizado e na vantagem de ser um software já validado.

OpenDSS pode ser utilizado na solução de problemas de simulação de redes de distribuição e transmissão de energia de qualquer tamanho, na análise de casos como:

- Fluxo de potência em redes balanceadas e desbalanceadas, monofásica, trifásica e multifásica;
- Simulações do tipo *Quasi-Static Time Series* (QSTS);
- Análise de faltas (curtos circuitos) inclusive utilizando a simulação de Monte Carlo;
- Análise harmônica e fluxo de potência harmônico;
- Análise de *flicker* ou cintilação para qualidade de energia elétrica;
- Análise dinâmica de sistemas eletromecânicos;

- Análise linear e não linear de sistemas;
- Modelagem de sistemas com correntes induzidas pelo campo magnético terrestre;
- Problemas de fluxo de potência e proteção em *smart-grid*.

OpenDSS possui uma biblioteca de 43 componentes, incluindo os tradicionais usados nos sistemas de energia e permite o uso de modelos personalizados. Tem incorporado modelos de dispositivos de controle para despacho de energia, controle de fase, inversores, controle de *tap* de transformadores, modelo de custos, sistemas de proteção e falhas. Os modelos estão documentados em (EPRI, 2012).

Nas versões recentes do OpenDSS incorporam-se ferramentas de alto desempenho como sistemas de processamento paralelo, elementos multi-agente, híbridos e de co-simulação. Disponibiliza uma linguagem de programação de rotinas (*scripts*) e inclui um *kit* de desenvolvimento para integração com outros aplicativos.

Em resumo, apresenta características de flexibilidade no uso e aplicação, possui um caráter sustentável pela atualização e desenvolvimento contínuo mediante contribuições de instituições de investigação e de usuários. O programa OpenDSS tem sido usado por estudantes e investigadores em suas publicações como se demonstra na revisão bibliográfica e representa uma plataforma de cálculo comprovada. As desvantagens de OpenDSS entre elas a dificuldade da linguagem de programação, a falta de uma interface gráfica e intuitiva, e a limitação de somente contar com suporte no sistema operacional Windows representando uma oportunidade para realização de este trabalho.

O objetivo deste trabalho consiste no desenvolvimento da interface gráfica DiagramaBarra apoiado no núcleo de cálculo de OpenDSS, conservando a compatibilidade e a validade dos modelos utilizados. As facilidades geradas no DiagramaBarra permite usar o software como ferramenta pedagógica, entregando ao professor e ao aprendiz uma ferramenta visual para realizar diagramas e simulações de sistemas de energia elétricos, com características flexíveis e multi-plataforma.

1.3 CONTRIBUIÇÕES DECORRENTES DESTE TRABALHO

No decorrer do desenvolvimento deste trabalho, foi publicado o seguinte artigo técnico:

PRIOSTE, F. B.; ALVAREZ, G. R.,
Impactos da Interligação de uma Usina Solar Fotovoltaica com
Controle Ativo de Tensão em um Sistema de Distribuição.
In: Simpósio de Especialistas em Planejamento da Operação e
Expansão Elétrica 2018 (SEPOPE), Recife, Pernambuco.

O software desenvolvido disponibilizou-se para descarga no site *SourceForge.net*² onde estão compartilhados os instaladores, códigos fontes, guias de instalação, guia do usuário e arquivos exemplos de diagramas.

² Site SourceForge <http://sourceforge.net/projects/diagramabarra> . Acesso em 10 de fevereiro 2019

2 SOFTWARE LIVRE E ENSINO NA ENGENHARIA ELÉTRICA

2.1 INTRODUÇÃO

Para conhecer o estado atual da tecnologia e sua relação com a educação, pesquisa-se neste capítulo as diferentes origens do software, como ele evoluiu até o estado atual, projetando o futuro a partir do ecossistema informático existente. Procura-se a aplicação da filosofia de software livre nas aplicações usadas nas instituições de ensino. Posteriormente, revisa-se a aplicação do *software* no ensino a partir de uma pesquisa bibliográfica.

Com a finalidade de relacionar a educação com o obtido acerca do software, revisam-se os diferentes paradigmas do conhecimento que dão espaço para o mecanismo de aprendizagem. Toma-se como referência o pensamento crítico-reflexivo e descreve-se a técnica de ensino baseado neste paradigma. Finalmente definem-se os requerimentos da plataforma de software para apoiar técnicas de ensino prático-reflexivos.

2.2 CONCEITO DE SOFTWARE LIVRE

Quando foram comercializados os primeiros computadores, eles não possuíam um sistema operacional ou sequer existiam linguagens de programação. Os dispositivos efetuavam sequências de instruções inseridas numa console ou mediante cartões perfurados. O processamento e o resultado da sequência eram entregues na forma de relatórios impressos por uma máquina de escrever elétrica. O *software* consistia numa série de instruções e comandos que o mecanismo computador devia realizar para obter o resultado.

Com o surgimento dos sistemas operacionais, o *software* passa a formar parte do pacote de processamento do computador. Nessa época aparecem as primeiras linguagens de programação que permitiram personalizar, ampliar o uso do computador e realizar mais funções. Como todo esse desenvolvimento acontecia no ambiente universitário, não existiam barreiras nem licenças ou taxas de direitos autorais, e os conhecimentos eram compartilhados entre pesquisadores. Porém as

empresas, que perceberam o potencial econômico na indústria do *software*, começaram a impor condições NDA (*Non-disclosure Agreement*) para proteger direitos autorais, resultando em séries de processos judiciais. O impacto negativo das ações legais motivou a comunidade de pesquisadores a refletir sobre o futuro do *software*, que ficaria sempre condicionado a grandes empresas como IBM e AT&T. Assim, nos anos 80 estabeleceram-se os pilares do que hoje chamamos *software* livre:

- O projeto GNU (*GNU is Not Unix*), cujo objetivo era desenvolver um sistema operacional que fosse *software* livre;
- A criação do FSF (*Free Software Foundation*), uma organização com o fim de proteger o livre desenvolvimento.
- A licença GNU GPL (*General Public License*), que garante que o *software* seja sempre livre de pagamentos.

Os princípios fundamentais do *software* livre estabelecem quatro liberdades.

Liberdade n°0

A liberdade de executar o programa para qualquer propósito

Liberdade n°1

A liberdade de estudar como o programa funciona e adaptá-lo para as suas necessidades

Liberdade n°2

A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo, sem ações ilegais.

Liberdade n°3

A liberdade de modificar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade seja beneficiada.

(OLIVA e MELO, 2009)

Desse movimento nascem os sistemas operativos Linux, o pacote de produtividade LibreOffice, o servidor web Apache, sendo este último o servidor web mais utilizado na internet³, além de muitas outras iniciativas de *software* livre. Com o

³ De acordo a informação coletada por <http://www.netcraft.com> . Acesso em 10 de fevereiro 2019

tempo algumas das características da licença GNU GPL foram adotadas por outras organizações que criaram licenças do tipo livre como *Creative Commons License* (CCL) e *Apache License*, para citar algumas. Hoje até grandes empresas de *software* como Microsoft e Oracle, disponibilizam versões livres de seus produtos comerciais usualmente chamados de "*Community Version*".

Hoje em dia, existe um amplo espectro de licenças do tipo *software* livre, combinando vários tipos de restrições de uso, distribuição, modificação, para uso privado, sendo mais permissivas ou menos permissivas, na medida das necessidades do desenvolvedor. Como referencia, se apresenta um extrato de tipos de restrições na Figura 1, onde pode ver-se que nas licenças *BSD Zero Clause License* e *Apache License 2.0* não disponibilizam o código fonte mas está permitido o uso comercial.

Figura 1 – Extrato da lista de tipos de licenças de software livre

License	Commercial use	Distribution	Modification	Patent use	Private use	Disclose source	License and copyright notice	Network use is distribution	Same license	State changes	Liability	Trademark use	Warranty
BSD Zero Clause License	●	●	●		●						●		●
Academic Free License v3.0	●	●	●	●	●		●			●	●	●	●
GNU Affero General Public License v3.0	●	●	●	●	●	●	●	●	●	●	●		●
Apache License 2.0	●	●	●	●	●		●			●	●	●	●

Fonte – <http://www.choosealicense.com/appendix> . Acesso em fevereiro 2019

As vantagens de contar com *software* livre são: uma maior participação de mercado, confiabilidade, estabilidade, segurança e um menor custo, usando-se a expressão TCO (*Total cost of ownership*) que considera não só o custo de licença, mas também o custo de treinamento, de manutenção e atualização de *hardware*. (OLIVA e MELO, 2009)

2.3 SOFTWARES LIVRES NO ENSINO.

Na área do ensino, são usadas ferramentas de *software* com propósitos administrativos e acadêmicos. Na UDESC a gestão dos recursos informáticos é realizada pela Secretaria de Tecnologia da Informação e Comunicação (SETIC), que tem por finalidade prover o planejamento, políticas, padrões e soluções de TI (Tecnologia da Informação). A SETIC têm por competências realizar a planificação e investimento no campo do *software* nos planos de governança, sistema administrativo, sistema acadêmico, portais corporativos, infraestrutura e suporte.

De acordo com os relatórios de gestão da SETIC de 2014 a 2017, uma parte do inventário de *software* e sistemas utilizados na UDESC corresponde a *software* livre (REITORIA, 2018;2017;2016;2015). O nível de investimento em licenciamento de *software* entre os anos 2013 e 2017 mostram-se na Tabela 1 onde se aprecia um constante incremento no ativo intangível a pesar da diminuição no investimento. Os valores não tem correção monetária e provêm do texto do relatório.

Tabela 1 – Investimento de licenciamento de *software* na UDESC

Ano	Locação de Bens Móveis, Imóveis, Equipamentos e Softwares	Total Aquisição de <i>Software</i>	Investimento desenvolvimento próprio	Ativo intangível (dados dos relatórios)
2017	1.736.749	398.245	Sem dados	Aumento 12%
2016**	2.432.457	226.418	Sem dados	Aumento 14%
2015*	1.878.660	399.171	1.314.152	Aumento 31%
2014	1.740.936	533.593	Sem dados	Aumento 60%
2013	1.310.493	388.200	Sem dados	Sem dados

Fonte: Produção do próprio autor a partir dos dados dos Relatórios de Gestão 2017, 2016, 2015 e 2014. (REITORIA, 2018)

No presente, a UDESC encontra-se realizando a implantação de um sistema de inventario de *software* centralizado, portanto ainda não existe um numero exato de quantos sistemas são de *software* livre e quantos são *softwares* comerciais. No caso particular do Centro de Ciências Tecnológicas (CCT) da UDESC, uma lista dos

softwares licenciados e livres disponibilizados para alunos e servidores está publicada no site da universidade⁴ conforme o mostrado na Tabela 2.

Tabela 2 – Lista de *Softwares* do centro CCT da UDESC

Tipo licenciamento	Nome <i>software</i>	
Licenciados para todas as estações	<ul style="list-style-type: none"> - Sistema operacional Microsoft Windows 7 e Windows 10 - Microsoft Office 365 - Antivirus Microsoft Security Essentials 	
<i>Software</i> livre	<ul style="list-style-type: none"> - BrOffice 3.0 - Adobe Reader - PDFCreator - WinDjView - Foxit Reader 	<ul style="list-style-type: none"> - CDBurnerXP - 7-Zip - Mozilla Firefox - Google Chrome - UltraVNC
<i>Softwares</i> com licença flutuante	<ul style="list-style-type: none"> - Abaqus - Altair - Altium - Ansoft Maxwell e Pexpert - Ansys - AutoDesk Autocad, Inventor e Revit 	<ul style="list-style-type: none"> - Labview - Maple 2015 - Matlab - MSC Adams, Easy5 e Marc - Orcad - Origin - Xinlix

Fonte: Dados do site² (CCT UDESC, 2018)

2.4 SOFTWARES DE SIMULAÇÃO DE REDES DE POTÊNCIA

A diminuição dos custos de sistemas computacionais nos últimos tempos provocou a expansão do mercado do *software*. Portanto existem diversas soluções de aplicativos para simulação de sistemas elétricos de energia. Na medida em que se desenvolvem tecnologias na área elétrica, surgem novas ferramentas para gerenciar, modelar e analisar o impacto da aplicação da inovação. O maior desenvolvimento ocorre nos centros de estudos. Entre os *softwares* de simulação de sistemas elétricos de energia citam-se os mostrados na Tabela 3.

⁴ <http://www.cct.udesc.br> . Acesso em fevereiro 2019.

Tabela 3 – *Software* de simulação de redes de potência

Nome	Licença	Nota
PowerWorld www.powerworld.com	Comercial	Versão gratuita limitado a 13 barras
ETAP etap.com	Comercial	
DigSILENT PowerFactory www.digsilent.de	Comercial	
Matlab + SimPowerSystems Toolbox www.mathworks.com	Comercial	Interface gráfica Simulink.
PSCAD/EMTP	Comercial	
CYME	Comercial	
Xandee	Comercial	Interface Web, usa OpenDSS para cálculos.
OpenModelica	Gratuito	Interface gráfica com blocos, código aberto
InterPSS	Gratuito	JAVA, interface gráfica, projeto sem código aberto
Octave + MATPower	Gratuito	Interface texto
PSAT	Gratuito	Requer Matlab + Simulink.
PYPOWER	Gratuito	Interface texto
GridSim	Gratuito	
OpenDSS	Gratuito	Interface texto, código aberto
GridLab-D	Gratuito	Interface texto
ATP/EMTP	Gratuito	Código fechado

Fonte: Produção do próprio autor a partir de informação disponível em https://wiki.openelectrical.org/index.php?title=Power_Systems_Analysis_Software

Ainda que algumas das soluções apresentadas sejam gratuitas, estão implementadas sobre uma plataforma de tipo comercial. Por exemplo *PSAT* e *MatPower* são gratuitos e livres, mas estão implementados no *Matlab*. Por outro lado *Octave* é uma plataforma gratuita similar a *Matlab*, porém mais limitada, pelo qual existem algumas incompatibilidades.

Alguns aplicativos adicionalmente requerem um intérprete instalado no computador, como por exemplo *InterPSS* e *PYPOWER*, que estão programados no linguagem computacional *JAVA* e *Python* respectivamente.

2.5 REVISÃO BIBLIOGRÁFICA

Para determinar o estado atual acerca do uso de *software* livre para simulação de redes de potência realizou-se uma pesquisa nos motores de busca IEEEXplore⁵ e arXiv.org⁶. Inicialmente definiram-se os seguintes critérios.

- a) Na primeira exploração definem-se as palavras de busca: *open source software, simulation software, power distribution, power grid, power system*;
- b) Definem-se o intervalo desde 2009 até 2016 agrupados por biênios. Esta pesquisa pode abranger períodos anteriores, porém é de esperar que vários resultados estejam já obsoletos. Define-se a categoria “Prévia a 2009” como referência;
- c) Possivelmente existem outros trabalhos similares realizados, portanto realizou-se uma busca específica por documentos de revisão bibliográfica incorporando as palavras de busca *review* e *survey*;
- d) Como critério de exclusão, descartaram-se algoritmos, simulações de casos particulares, exemplos de textos e otimizações;
- e) Como critério de inclusão, incluem-se estudos comparativos, avaliações e estudos de rendimento.

Na pesquisa realizada, usou-se a classe de intervalo de tempo mais recente para conseguir no máximo 100 documentos, considerando a posterior tarefa de leitura dos documentos. Na primeira etapa de filtragem, se classificaram os documentos pelo conteúdo do resumo e das palavras chaves comuns. Uma segunda etapa de filtragem, reuniu-se a informação das aplicações mais comuns para uso de *software*. Na busca efetuada foram encontradas revisões bibliográficas, não sendo necessário aplicar o segundo filtro.

Na Tabela 4 apresentam-se os resultados da busca sistemática no IEEEXplore. Na ferramenta de busca de arXiv.org foram encontrados cinco documentos de data mais recente relacionados com engenharia elétrica e nenhuma revisão sistemática.

⁵ <http://www.ieeexplore.org> . Acesso em fevereiro 2019

⁶ <http://www.arxiv.org> . Acesso em fevereiro 2019

Tabela 4 – Resultados de busca bibliográfica

Busca	Ant. 2009	2009-10	2011-12	2013-14	2015-16
<i>Open source</i> <i>+ simulation + software</i>	414	111	228	234	259
<i>Open source + simulation</i> <i>+ software + power</i>	89	24	36	37	37

Fonte: Produção do próprio autor

Uma vez que se obtiveram a quantidade prevista de resultados para um primeiro nível de filtragem, realizou-se uma leitura dos resumos e registro de palavras chaves. Nesta etapa foram descartados todos os trabalhos específicos referentes a simulações de problemas particulares, análises, exemplos de textos, otimizações, algoritmos e estudos fora do escopo de sistemas de energia. Nos conteúdos dos artigos, os conceitos relacionados com *software* mais citados foram *PSAT*, *OpenDSS*, *Python*, *PyPower*, *MatPower*, *GridLAB-D*, *Java*, *Matlab*, *Simulink*, *C++*, *smartgrid*, *co-simulation*, *Real-Time*, *System-on-a-Chip*, *PowerFactory*. Nos resultados de desenvolvimentos mais recentes encontraram-se avaliações do *software e-Asimov* (HARTMANN, ÇAKMAK, *et al.*, 2017) e *OpenModelica* (ZHANGBIAO e SHANAN, 2013) utilizados como sistemas multi-agente.

Nesta busca somente um resultado contém a palavra chave *review* com conteúdo fora do escopo do presente trabalho pelo critério de exclusão. Em vista do resultado, realizou-se uma nova busca omitindo o termo *open source* encontrando-se 5 revisões resumidas na sequência.

O trabalho publicado por Milano e Vanfretti (2009) apresenta uma detalhada revisão de ferramentas de simulação de *software* livre. O texto descreve definições de *software* proprietários, *software* gratuito, e *software* de código aberto e as diferentes modalidades de licenciamento. Os autores dividem o objetivo de uso das aplicações na educação/investigação/desenvolvimento e aplicação nos negócios. Logo se analisam 10 aplicativos livres para análise de sistemas de energia: *UWPFLOW*, *TEFTS*, *MatPower*, *PSAT*, *VST*, *InterPSS*, *AMES*, *DCOPFJ*, *Pylon*, *OpenDSS*. Destacam-se as vantagens de utilizar estes aplicativos na educação e investigação por serem soluções de código aberto permitindo a modificação. Cita-se a possibilidade de negócios vinculados com o desenvolvimento do *software* de simulação.

A publicação de Kondoro et al (2017) realiza a aplicação de vários *softwares* na simulação de *smart-grid*, avaliando diferentes aspectos no desenvolvimento de um problema. Realiza a análise dos aplicativos *AnyLogic*, *Repast*, *RAPSim* e *GridLAB-D*. No documento discute-se a modelagem de painéis fotovoltaicos na simulação de sistemas de energia usando dados climáticos concluindo que ferramentas de propósito geral cumprem etapas iniciais de estudo, mas somente com integração de diferentes ferramentas consegue-se o objetivo de analisar comunicações e controle de redes *smart-grid*.

No estudo de Porchacker (2013) lista-se a funcionalidade dos aplicativos livres *UWPFLOW*, *TEFTS*, *MatPower*, *PSAT*, *IPSYS*, *MatDyn*, *AMES*, *InterPSS*, *OpenDSS*, *GridLab-D* nos distintos tipos de análise. Posteriormente, avalia-se o desempenho de *MatPower*, *PSAT*, *InterPSS* e *GridLAB-D* na solução do problema de 14 barras da IEEE⁷. O estudo verifica a possibilidade de usar ferramentas na simulação de *smart-grid* empregando um modelo personalizado para incluir funções que não estão integradas no aplicativo, como modelos de mercado, econometria e análise de dados.

O uso de ferramentas de simulação fora do âmbito acadêmico sucede nas instituições de governo. A revisão preparada por Moffet, Sirois e Beauvais (Julho,2011) para o governo de Canadá no contexto do *Program on Energy Research and Development* (PERD) analisa o impacto das novas tecnologias de geração de energia distribuída, veículos elétricos e *smart-grid*. Na revisão descrevem-se em detalhe os aplicativos *OpenDSS*, *GridLAB-D* e *APREM*. Apresentam-se os tipos de simulações que podem ser realizados, funcionalidades, limitações, documentação, requerimentos e disponibilidade de exemplos. Utiliza-se um sistema de geração distribuída com 3 geradores eólicos, com mudanças topológicas de rede considerando variações de carga e vento com intervalo de simulação de um ano discretizados de uma em uma hora. Avalia-se nas simulações os cálculos das perdas elétricas no sistema ao longo do ano e o tempo requerido para realizar as simulações.

A partir da leitura das revisões encontradas pode-se concluir que no tema de simuladores de sistemas de energia existem soluções dependentes do objetivo de análise. Distinguem-se duas aplicações principais: 1) análises no domínio do tempo:

⁷ Disponível em <http://labs.ece.uw.edu/pstca/> . Acesso em fevereiro 2019

que correspondem as análises de fenômenos transitórios eletromecânicos, transitórios eletromagnéticos e fenômenos de onda; 2) análises no domínio da frequência: correspondem as análises de regime permanente e as análises harmônicas. Diferencia-se a capacidade de análise nos esquemas de transmissão AC, DC, HVAC e HVDC.

Aplicações de geração distribuída e *smartgrid* têm uma crescente demanda, surgindo interesse nos métodos de: simulação em sistemas distribuídos, que são realizados em *clusters* de processamento; simulação híbrida, onde uma parte da matemática é emulada por *hardware* do tipo FPGA ou SoaC (*System-on-a-Chip*); co-simulação (STIFTER, WIDL, *et al.*, 2013) e sistemas multi-agentes (CHOUHAN, 2009), que consideram a parte estrutural matemática de transmissão de energia e problemas estocásticos na transmissão de dados dos sistemas de controle.

No âmbito educacional, as publicações encontradas (POYRAZOGLU e OH, 2016) e (TURTIAINEN, MANNILA, *et al.*, 2002) focam-se no estudo de estratégias e simulações de econometria dos sistemas de distribuição de energia elétrica.

Revela-se que várias soluções para simulação apoiam-se no *software* OpenDSS e GridLAB. Como referência, OpenDSS foi citado em 229 publicações e GridLAB em 97 publicações usando uma busca no IEEEXplore no intervalo de tempo definido. Existem alguns trabalhos focados na construção de uma interface gráfica usando *software* SCADA tal como a publicação de Montenegro e Dugan (2017), mas estão condicionadas ao uso de *software* comercial licenciado.

2.6 PARADIGMAS E APRENDIZAGEM

O ensino consiste na transmissão do conhecimento de uma pessoa (tutor) a outra (aprendiz), e representa uma das bases da cultura humana. No ocidente, se diz que a educação foi estudada inicialmente como fenômeno social pelos filósofos gregos, chamaram-na de didática, que etimologicamente significa no idioma grego “a arte de ensinar”. Somente na época da Ilustração do século XVIII, com Descartes e Kant, o racionalismo permite estabelecer o processo educativo como ciência e são construídas as primeiras metodologias pedagógicas (LODER, 2009). Durante o século XX, motivados pelas revoluções sociais e a aparição de novos paradigmas da ciência, filósofos, sociólogos, psicólogos e estudiosos sistematizaram o processo

educativo. Surgem educadores de renomes como Piaget, Freire, Vygotsky, Feuerstein, e muitos outros. O trabalho de Moreira e Massoni (2011) mostra que a partir das teorias de construção do conhecimento surgem diversos modelos educativos e representações do que consiste o ensino e a aprendizagem.

Considera-se que no ensino clássico o maestro expunha a matéria e o aprendiz adquiria habilidades e conhecimento mediante a repetição de autores e obras, de forma dogmática e mecânica. No caso que o aprendiz cometesse erros ou questionasse o fundamento dos autores, ele era penalizado chegando a usar-se o castigo físico e a segregação social. Ainda nos dias de hoje existem abusos desse tipo em certas culturas. Porém, é uma metodologia pouco eficiente que não satisfaz a necessidade de massificar o conhecimento. (ARANHA, 1996)

Para transmitir o conhecimento, é necessário saber dos mecanismos de como este se constrói. A epistemologia é a parte da filosofia que estuda os raciocínios que levam a criação do conhecimento. Na Tabela 5 apresenta-se uma breve resenha das linhas epistemológicas recentes, sua descrição e autor.

Tabela 5 – Principais linhas epistemológicas contemporâneas

Denominação e autor	Descrição
Histórica de Bachelard	A ciência, e por analogia, o conhecimento, nasce e evolui a partir do aprendido historicamente, importando a origem, estrutura e evolução do conhecimento científico.
Arqueológica de Foucault	Extensão da Histórica, pondo o ser humano no centro da história científica, dependendo do lugar, cultura e período de tempo quando ocorre.
Genética de Piaget	O conhecimento é uma construção intelectual produto da interação entre sujeito e ambiente. O progresso no indivíduo se realiza mediante o método psicogenético. Trata-se de um processo aberto e contínuo, resultado de reações do sujeito perante a atividade prática e mental.
Racionalista Crítica de Popper	A ciência consta de um conjunto de teorias comprovadas pela verificação experimental. O método científico consiste no processo de verificação de cada fundamento.
Crítica de Habermas	Sugere que a ciência nasce do interesse por descobrir a natureza das coisas, a partir da qual surge a construção de teorias, e na busca da origem, propõe experimentos para a comprovação crítica. Portanto existe um pre-conhecimento, a estruturação do experimento, o evento, a interpretação do resultado e a posterior conclusão. Estruturação e interpretação dependentes do tempo, sendo subjetivas.

Denominação e autor	Descrição
Descritiva de Kuhn	Estabelece que a ciência depende do paradigma ou perspectiva sobre a qual se analisa o problema e trata de resolver. Esse paradigma tem limitações dependendo da cultura e do tempo. Cada vez que um paradigma perde validade pela acumulação de erros e incongruências, surge um novo paradigma, produzindo uma revolução científica. O conhecimento, por conclusão, consiste numa sequência de revoluções científicas.
Biológica de Maturana	Propõe que o conhecimento depende do observador, e como observador tem interpretação própria. Mediante o processo cognitivo, são geradas as estruturas lógicas que o sustentam o conhecimento num domínio particular de validade. A transmissão do conhecimento depende da linguagem, e representa uma tentativa do observador reformular o adquirido em um léxico comum. A ciência consta de um conjunto de conhecimentos regidos por critérios estabelecidos de validação e que contém seu próprio léxico.

Fonte: Produção do próprio autor a partir de (MOREIRA; MASSONI, 2011; LODER, 2009; CANTO; MÜLLER; LIMA, 2015).

Porém nos paradigmas que fundam as epistemologias descritas, onde o processo educativo se atomiza e sistematiza, se desconsideram fatores exógenos que influenciam a construção individual do conhecimento, como são a motivação, a integração cognitivo-afetiva, e os fatores étnico-culturais. A presença de culturas asiáticas, indígenas e afro na sociedade multicultural existente, impede que o racionalismo ocidentalizado seja aplicado de forma efetiva em todo âmbito educativo. Nesse contexto sobressai o trabalho de Maturana (2001) e Dussel (1980), ambos propõem uma perspectiva holística do conhecimento humano, incorporando outras dimensões e incluindo a cosmovisão latino-americana.

Na visão biológica de Maturana (2001), no processo educativo existe uma relação humana já que tanto o aprendiz como o tutor são seres humanos completos, que possuem sentimentos, histórias e valores individuais, assim como estruturas mentais particulares e capacidade de adaptação (ROSSETTO, 2008). A fim de favorecer a tarefa educativa, deve-se contemplar a conquista do aprendiz, estimular seu interesse e dar espaços para gerar canais de comunicação, em certos casos modificar a estrutura social predeterminada. O trabalho de análise de Bazzo (2012), descreve este fenômeno no âmbito da educação superior, onde o aprendiz constrói vias de comunicação fora do contexto disciplinar para produzir um contexto

educativo informal, de modo a favorecer um objetivo, seja este a aprendizagem ou uma melhor qualificação.

Dussel (1980) apresenta uma visão originária do ser latino-americano, conquistado e explorado, que não deixou ser domesticado, que ainda têm raízes pré-colombianas vivas no cotidiano. O indivíduo, geralmente produto de carências extremas, abusos, desigualdades e danificado socialmente, requer de uma projeção simbólica libertaria como eixo motivador, porque a recompensa moral pelo esforço não tem equivalência nas crenças do povo nativo. A educação branca e católica se impõe à força pelo conquistador. Uma realidade educativa universal deve ser elaborada a partir da desconstrução do rol comandante-comandado. Esta deve fundar-se sobre um convênio de mútua cooperação entre dois seres humano com metas individuais e coletivas dentro da sociedade. Um exemplo do paradigma Dusseliano existe na Bolívia, onde a educação formal tem currículo flexibilizado com idiomas nativos em uma política educativa descolonizadora (LENZA, 2015).

Neste contexto de variadas epistemologias e paradigmas, o maestro pode usar diferentes ferramentas metodológicas para ensinar ao aprendiz. O processo educativo torna-se um processo racionalista, empírico e reflexivo. De acordo com o texto de Diaz Bordenave e Pereira (2015), numa visão sistêmica do processo de aprendizagem podem identificar-se diferentes formas na qual o conhecimento é adquirido. O conhecimento é adquirido de forma direta utilizando:

- Memorização mediante repetição, exercícios e regras mnemotécnicas;
- De modo cognitivo, utilizando a lógica e a razão;
- De modo compensativo, mediante analogias, mímicas e suposições.

Também se adquire conhecimento de forma indireta através de mecanismos:

- Meta-cognitivos, através da linguagem;
- Afetivos, através de atividades prazerosas e atrativas para o aprendiz;
- Sociais, usando conhecimentos na atividade cultural e social.

O aprendiz tem um rol mais ativo que a simples repetição mecânica. As atividades educativas orientam para trabalho do tipo individual, do tipo grupal com os

parceiros, do tipo autoaprendizagem, realizados fora da aula, e do tipo auto-metas, com objetivos fora da matéria da aula.

Na visão sistemática o tutor tem três tarefas

- Ser a autoridade que apresenta as regras de avaliação;
- Ser o elemento de controle que avalia e apresenta os prazos;
- Ser o elemento de coordenação do processo.

“A participação do professor nesse processo consiste em planejar estratégias pedagógicas que, em um primeiro momento, possam fomentar o interesse do aluno pela matéria e, na sequência, possam servir de suporte para que esse processo de aprendizagem seja bem sucedido” (LODER, 2009, p. 155).

Loder (2002) afirma que a engenharia corresponde a um construtivismo empírico. A epistemologia aplicada do professor de engenharia mostra um empirismo marcadamente positivista, com falta de preparo pedagógico, produto do constante aperfeiçoamento técnico. Portanto, o desempenho na docência se correlaciona na medida em que o professor acumula experiência, sendo dominante o uso de uma metodologia pedagógica diretiva. Bazzo (2012) evidencia essa disposição dentro das faculdades de engenharia, atribuindo-se ao fato de que o corpo docente possui formação em engenharia e não uma formação pedagógica.

Por outro lado, no texto de Anderson e Lawton (2009), cita como referência a classificação de aprendizagem do domínio cognitivo que ocorrem quando são usadas ferramentas de simulação. Ainda que o trabalho citado refere-se a comparar o aprendizado concreto com a *percepção* de aprendizado, realiza um modelo a partir da classificação tradicional de Bloom usada no ensino. No novo contexto proposto, a tarefa dirigida passa a ser uma tarefa autônoma.

2.7 METODOLOGIAS PARA O ENSINO PRÁTICO-REFLEXIVAS.

A partir dos paradigmas da ciência descritos, destaca-se no ensino a aplicação da epistemologia de Piaget (1983) e o pensamento crítico como linha de pensamento para um melhoramento continuado do processo educativo.

Como difusor da cultura reflexiva, Schön (2000) desenvolve as noções de que tanto as pesquisas como o trabalho prático permitem melhorar seu desempenho

da atuação profissional. Ele tem que adaptar-se na medida em que surgem novas ideias, instrumentos e tecnologias. No contexto educativo, dependendo do perfil de atividades realizadas pelo aprendiz o conhecimento é adquirido por diferentes mecanismos. No esquema pratico-reflexivo, Schön identifica duas instâncias de aprendizado que denomina conhecer-na-ação e reflexão-na-ação.

Na atividade da aula, a matéria apresentada tem um exemplo, e na medida em que se desenvolve o exercício, a teoria se correlaciona com a aplicação. Este processo é cognitivo a nível primário, sem fixação no longo prazo, porém a sequencia lógica pode ser lembrada e repetida. Conhecer-na-ação ocorre quando o aprendiz realiza o exercício de forma autônoma de forma individual ou grupal. Na repetição o processo passa a ser cognitivo de nível secundário e passa para o processo de memorização. Parte da exercitação na disciplina matemática funciona dessa forma, sendo a metodologia aplicada nos textos de estudo da série Schaum.

Reflexão-na-ação é uma reconstrução mental retrospectiva, estimulada posteriormente de uma série de exercícios, com a finalidade de fixar os conhecimentos adquiridos. Processo que pode ser dirigido ou espontâneo.

Schön também cita a reflexão-sobre-a-reflexão-na-ação como um ato recursivo realizado quando o aprendiz faz uma descrição da reflexão-na-ação do exercício, de modo tal a transmitir o aprendido.

Como descrito na Seção 2.6 identifica-se a metodologia diretiva do professor, deixando ao aprendiz o trabalho de cumprir tarefas para conseguir uma qualificação. Porém, aspecto como autonomia, iniciativa e trabalho grupal não são avaliados. Com o objetivo de incorporar competências na formação do engenheiro eletricitista, propõe-se aplicar temas transversais nas atividades propostas no Apêndice B. Considera-se tema transversal, toda aquisição de experiências qualitativas alheias ao objetivo principal acadêmico e que são incorporadas de forma indireta. Segundo o Ministério da Educação (MEC), “são temas que estão voltados para a compreensão e para a construção da realidade social e dos direitos e responsabilidades relacionados com a vida pessoal e coletiva e com a afirmação do princípio da participação política” (BRASIL,1999). Exemplos de competências transversais dos profissionais são: a iniciativa, a flexibilidade, a noção de planejamento, a aptidão para gerir. Outras competências transversais relacionam o individuo com a sociedade, como é a ética, saúde, meio ambiente, pluralidade e cultura.

Nos temas transversais para a aplicação atual consideram-se relevantes algumas dimensões como crescimento pessoal, aspectos cognitivos, formação ética, entorno pessoal e tecnologias da informação⁸.

Sendo assim, definiram-se os seguintes temas transversais.

- Favorecer a autonomia no campo cognitivo para tomar de decisões;
- Estimular a iniciativa para propor de soluções a problemas;
- Delegar e repartir responsabilidades entre pares;
- Aumentar o grau de confiança em si mesmo e entre pares;
- Adquirir critérios qualitativos a partir da experiência na interface;
- Limitar a duração do tempo para resolver problemas;
- Promover o uso de ferramentas de busca de informação técnica;
- Sistematizar o ciclo de trabalho no uso de programas de simulação.

Incorporando os temas transversais na metodologia pedagógica, permitirá o aprendiz adquirir conhecimentos apoiados no compartilhamento e na experiência, diminuindo tempos nas tarefas de estudo.

2.8 ESPECIFICAÇÕES E REQUISITOS DA PLATAFORMA

A partir da análise realizada nas secções anteriores, pode-se concluir que uma plataforma de simulação aplicada no ensino deveria ter as seguintes características:

- Seguir os critérios do pensamento crítico-reflexivo;
- Permitir realizar atividades dirigidas e autônomas;
- Ter um suporte continuado durante a disciplina ou por várias disciplinas;
- Incorporar relações simbólicas entre objetos cognitivos.

Usando a classificação de Anderson e Lawton (2009), os objetivos do uso da plataforma devem permitir.

- Criar problemas e exercícios;

⁸ Baseado nos recursos de apoio do MEC - SEF. Parâmetros Curriculares Nacionais

- Avaliar a os resultados de tarefas conferindo resultados com o grupo de estudo;
- Analisar resultados e descrever a solução dos exercícios;
- Aplicar o aprendido em um exemplo do mundo real;
- Entender e estabelecer relações entre componentes do exercício.
- Lembrar os conhecimentos adquiridos mediante uma auto avaliação.

O resultado deste trabalho pretende ser um apoio na compreensão da matéria de disciplinas do curso de Engenharia Elétrica e ser uma ferramenta para uso na futura vida profissional do aprendiz. Também se espera que permita ao usuário realizar contribuições na interface.

Dentro dos requisitos técnicos para entregar uma plataforma de simulação de sistemas de energia elétrica para aplicação no ensino é essencial que siga certas linhas para permitir a difusão e sustentabilidade no tempo. Como primeiro requisito está possuir características de portabilidade dentro dos atuais sistemas operacionais usados em computadores e *tablets*: *Microsoft Windows*, *Linux* e *Mac OS*. Para que haja possibilidades de sustentabilidade no tempo, a plataforma de programação escolhida deve contar com atualizações frequentes e suporte continuado.

Para favorecer a usabilidade e universalidade, são desejáveis as seguintes características.

1. Permitir o usuário modificar idioma interface;
2. Interface gráfica e simbólica;
3. Sistema de armazenamento de arquivos compatível com XML (eXtensible Markup Language);
4. Incorporação de componentes de bibliotecas externas;
5. Parametrização simples com ferramentas de ajuda;
6. Uso de uma ferramenta para criação de curvas paramétricas.
7. Seguir princípios de *software* livre, sob Licença Atribuição-Não Comercial-Sem Derivações 4.0 *Internacional* CCL.⁹

As funcionalidades da interface a entregar de acordo a ementas do curso de engenharia elétrica serão no principio:

⁹ Disponível em <http://creativecommons.org/licenses/by-nc-nd/4.0/> , Acesso em fevereiro 2019

- Realizar diagrama unifilar;
- Obtenção de matrizes de admitância e de impedância;
- Executar simulações de fluxo de potência monofásicas e polifásicas balanceados ou não;
- Análise de perdas e impactos da inserção de novas fontes de energia no sistema;
- Armazenamento de energia via bancos de baterias e inserção de carros elétricos;
- Realizar gráficos no tempo a partir da mudança de condições operativas;
- Determinar limites de estabilidade estática;
- Cálculo de correntes de curto-circuito e das contribuições das novas fontes de energia para tais corrente.

Os tipos de atividades recomendadas para aplicação de exercícios nas tarefas para os alunos:

- Análise e compreensão dos resultados;
- Inferência acerca da importância dos parâmetros;
- Projeção dos efeitos no caso de mudanças topológicas e variações de geração e carga;
- Análise e correções de possíveis problemas que afetam o desempenho de redes elétricas.

3 DESENVOLVIMENTO DE UMA FERRAMENTA GRÁFICA DE SIMULAÇÃO

3.1 INTRODUÇÃO

Neste capítulo, será apresentado o trabalho de programação realizado. O programa desenvolvido tem por nome DiagramaBarra para indicar sua função na diagramação de esquemas de sistemas de energia.

3.2 FERRAMENTAS DE DESENVOLVIMENTO

Utilizou-se para o desenvolvimento de DiagramaBarra a plataforma Microsoft Visual Studio¹⁰, sendo desenvolvido na linguagem de programação C# usando o NET *Framework*¹¹. Os códigos fontes são compatíveis com a plataforma MonoDevelop¹², sendo possível também usar o programa no sistema operacional Linux e Mac. A extensibilidade de esta plataforma permite gerar programas para o sistema operacional Android, porém não será explorado neste trabalho.

A plataforma .NET Framework fornece a fundação para construir e executar aplicações dessa tecnologia. A plataforma dispõe de dois componentes, o tempo de execução de linguagem comum (*Common Language Runtime* - CLR) e as bibliotecas de categoria .NET (*Framework Class Library* – FCL). O CLR fornece gerenciamento de memória, controle de exceção, interoperabilidade, manipulação de processos simultâneos, reflexão, segurança e serviços de compilação para a arquitetura do sistema operacional. A FCL oferece biblioteca de objetos reutilizáveis para funções de acesso a arquivos, conectividade com banco de dados, redes, web, criptografia, acesso aos serviços do sistema operacional, estruturas de dados, ferramentas para suporte de várias linguagens e muito mais.

Entre os benefícios de usar NET *Framework* estão o rápido desenvolvimento ao dispor de uma ampla variedade de bibliotecas e recursos; maior

¹⁰ Disponível para descarga no site de Microsoft www.visualstudio.com

¹¹ <https://dotnet.microsoft.com/> . Acesso em fevereiro 2019

¹² Disponível em <http://www.mono-project.com> . Acesso em fevereiro 2019

confiabilidade e segurança no tempo de execução com o ambiente gerenciado e o isolamento de aplicativos, eliminando perdas de memória, violações de acesso e problemas de versão; esta baseada em padrões Web incorporando tecnologia unificada XML; acesso a bancos de dados e de conectividade; todos os componentes criados são tratados como objetos com propriedades, métodos, eventos e atributos, podendo ser importados. Atualmente, a última versão do NET Framework esta disponível para vários sistemas operacionais e possui código aberto (HICKSON, 2005).

Para facilitar o desenvolvimento do DiagramaBarra pesquisou-se a disponibilidade de bibliotecas ou *toolkits* com código livre que outorgassem características e funcionalidades específicas. Identificaram-se três funções independentes: interface gráfica para manipulação de diagramas, geração de gráficos e curvas, e processamento matemático para simulação.

Para manipulação de diagramas de forma gráfica foram encontrados as bibliotecas *OpenDiagram*¹³, *NShape*¹⁴, *FlowSharp*¹⁵ e *Diagram.NET*¹⁶. Cada biblioteca foi testada com uma interface simplificada que tivesse a possibilidade de personalizar ícones, eventos, conexões e vínculo a uma classe. Na Tabela 6 mostra-se um resumo tabulado dos testes realizados.

Tabela 6 – Comparação de bibliotecas para manipulação de diagramas

Características	OpenDiagram	NShape	FlowSharp	Diagram.NET
Ultima versão	5 (2010)	2.2.1 (2015)	1.0 (2016)	0.4 (2014)
Conexões e Drag&Drop	Sim	Sim	Sim	Sim
Personalização de gráficos	Não	Sim	Sim	Sim
Personalização de eventos	Sim	Não	Não	Não
Personalização de conexões	Não	Sim	Não	Não
Vínculo imagem a objeto	Não	Não	Sim	Não
Operação condicional	Não	Não	Sim	Não
Documentação	Não	Sim	Sim	Não

Fonte: Produção do próprio autor

¹³ Disponível em <https://archive.codeplex.com/?p=opendiagram> .Acessado em 10 de fev, de 2019

¹⁴ Disponível em <https://github.com/dataweb-GmbH/NShape> .Acessado em 10 de fev. de 2019

¹⁵ Disponível em <https://github.com/cliftonm/FlowSharp> .Acessado em 10 de fev. de 2019

¹⁶ Disponível em <https://github.com/dalsssoft/diagramnet> . Acessado em 10 de fev. de 2019

Durante os testes, conclui-se que a pesar da facilidade de gerar um editor gráfico, nenhum pacote cumpriu todas as funcionalidades requeridas. Além da falta de funções também foram notadas faltas de documentação, portanto optou-se por gerar uma nova biblioteca para diagramas que fosse simplificada para cumprir os requisitos de desenvolvimento do programa. Os códigos fontes dos pacotes na Tabela 6 serviram de referência para brindar algumas características. O desenvolvimento alcançado na nova biblioteca permite controlar aparência, ações e propriedades vinculadas com os ícones, realizar conexões condicionais e mostrar resultados diretamente no diagrama. A filosofia de *software* livre na qual esta baseado DiagramaBarra permitirá adicionar novas funções no futuro.

Para cumprir a função de geração de gráficos e curvas avaliaram-se as bibliotecas de código aberto *ZedGraph*¹⁷, *NPlot*¹⁸, *WebChart*¹⁹ e *MSCharts*²⁰. Ao testar os pacotes nota-se que *NPlot* e *WebChart* apresentaram abandono de suporte e perda de compatibilidade com as novas versões do NET *Framework*. O pacote *MSCharts* possui uma ampla documentação e com muitos exemplos, porém tem uma licença com distribuição restrita. *ZedGraph* foi considerada a melhor opção a ser utilizada por contar com documentação, suporte multi-idioma, e possuir licença de livre distribuição.

Adotou-se a biblioteca *OpenDSSDirect* para efetuar a função de processamento matemático. Esta biblioteca é fornecida como componente do simulador OpenDSS por parte do EPRI (*Electric Power Research Institute*) e seu código fonte está realizado na linguagem de programação Delphi estando disponível para descarga através do portal SourceForge.

OpenDSS e seus subcomponentes se distribuem baixo licença EPRI de 3 pontos similar a BSD onde “o uso e distribuição de códigos fontes e binários são permitidos: distribuição de códigos fontes mantendo os direitos autorais incorporando a licença EPRI nos códigos fonte, distribuição de binários mantenha os direitos incorporando a licença EPRI, e que o nome de EPRI ou colaboradores

¹⁷ Disponível em <https://github.com/ZedGraph/ZedGraph>. Acessado em 10 de fev, de 2019

¹⁸ Disponível em <http://netcontrols.org/nplot/wiki/index.php> .Acessado em 10 de fev, de 2019

¹⁹ Disponível em <http://www.carlosag.net/Tools/WebChart/> .Acessado em 10 de fev, de 2019

²⁰ Disponível em <https://code.msdn.microsoft.com/mschart> .Acessado em 10 de fev, de 2019

podem ser usados para promover produtos derivados de este software sem autorização escrita”²¹.

A integração da biblioteca com o programa DiagramaBarra foi realizada por chamadas a bibliotecas dinâmicas (*Dynamic link library*, DLL). Esta biblioteca contém os elementos de simulação, funções de manipulação de elementos e funções para recuperar os resultados da simulação. Possui capacidade para realizar cálculos de fluxo de potência monofásicos e polifásicos, pode efetuar análise de transitórios eletromecânicos, contém elementos para estudos de correntes induzidas pelo campo magnético terrestre, elementos de controle e possibilidade de usar modelos personalizados de carga e geração. OpenDSS está indicado pela Agência Nacional de Energia Elétrica – ANEEL – para cálculos de perdas na distribuição de acordo a notas técnicas (ANEEL, 2014) além de ser utilizado amplamente por investigadores e estudantes da área elétrica em diversos países.

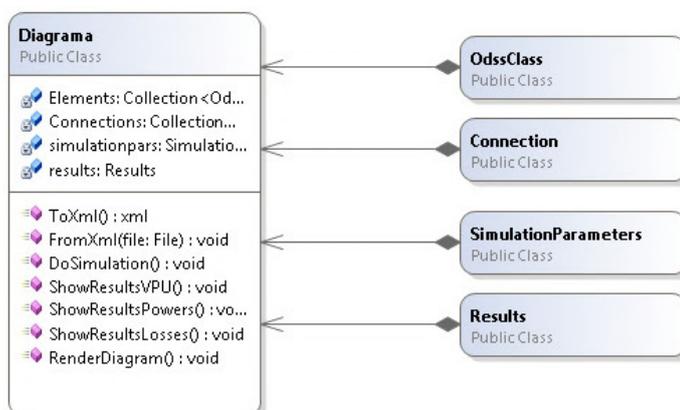
Na sequência descreve-se o modo de funcionamento, o modelo lógico e de estrutura de dados, posteriormente descreve-se o trabalho de programação da interface gráfica, para finalizar com o procedimento de simulação e apresentação do produto versão alpha.

3.3 DESENVOLVIMENTO DO MODELO DE DADOS

Um problema de sistema de energia possui um conjunto de elementos e um conjunto de conexões entre elementos. Seguindo esse conceito, o modelo de informação conterà uma coleção de elementos e uma coleção de conexões. Logo existem condições definidas para a análise do sistema, que se toma como parâmetros de simulação. Além do estado, se requer armazenar os resultados da simulação. A Figura 2 mostra o esquema resultante, o qual também servirá como base para o formato de arquivo.

²¹ Tradução livre do autor

Figura 2 – Estrutura classe *Diagrama* e armazenamento lógico



Fonte: Produção do próprio autor

A classe *Diagrama* contém a representação lógica do sistema de energia. Nesta classe programaram-se métodos para interagir com o armazenamento de arquivos *ToXml()* e *FromXml()*, para executar o cálculo do problema *DoSimulation()* e realizar a apresentação gráfica na interface *RenderDiagram()*, *ShowResultsVPU()*, *ShowResultsPowers()* e *ShowResultsLosses()*.

A continuação revisam-se as classes presente no esquema da Figura 2.

3.3.1 O elemento *ODSSClass*

Criou-se como uma classe a partir da qual se derivam todos os elementos da biblioteca OpenDSSDirect. O modelo da classe *ODSSClass* considera atributos gerais comuns para todos os elementos, como a posição no diagrama (*Location*), categoria (*CatId* e *FullId*), número de conexões (*nrobus*) e ícone (*Image*). Considera outros atributos auxiliares como vínculo a outro objeto (*Tag*) e sufixo para elementos monofásicos (*suffix*). Programaram-se métodos gerais para acesso a parâmetros (*getStrParValue()* e *getDoubleParValue()*), para manipulação do nome (*getName()* e *setName()*), para entregar comando de chamada (*getCommand()* e *toCommand()*), e manipulação XML para armazenamento (*fromXml()* e *toXml()*). Outro método particular programado permite determinar se o elemento tem arquivos vinculados (*hasFiles()* e *getFiles()*). A Figura 3 resume o descrito.

Figura 3 – Modelo da classe *ODSSClass*

Fonte: Produção do próprio autor

A partir da classe *ODSSClass* são derivados todos os elementos importados da biblioteca *OpenDSSDirect* com as propriedades individuais adicionadas. Para formalizar cada classe foi realizado um estudo das propriedades e funções de todos os componentes. Para conservar a compatibilidade foi replicada a estrutura a partir dos códigos fontes do *OpenDSSDirect*. A Tabela 7 mostra a tipologia dos elementos gerados, as funções de manipulação disponíveis e o ícone desenhado para ser usado na interface. Os ícones foram gerados imitando a simbologia utilizada nos textos de estudo e convenções.

Tabela 7 – Elementos da biblioteca *OpenDSS* incorporados ao *DiagramaBarra*

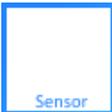
Tipo	Nome elemento	Id	Numero de conexões	Funções	Ícone
-	BusBar"	0	N	S, V	
Source/Fault	VSource"	1	1	I, S, F,V	

Tipo	Nome elemento	Id	Numero de conexões	Funções	Ícone
	ISource	2	1	I, S, F,V	
	Fault	3	1	-	
PDelements	Capacitor	4	1	I, S, F,V	
	GICTransformer ²²	5	0	-	
	Line	6	2	I, S, F,V	
	Reactor	7	2	-	
	Transformer	8	2 ou mais	I, S, F,V	
PCElements	Generator	9	1	I, S, F,V	
	GICLine ²³	10	1	-	
	Load	11	1	I, S, F,V	
	PVSystem	12	1	I, S, F,V	
	Storage	13	1	-	

²² Modelo de transformador para estudo de correntes induzidas pelo campo magnético da Terra.

²³ Modelo de linha para estudo de correntes induzidas pelo campo magnético da Terra

Tipo	Nome elemento	Id	Numero de conexões	Funções	Ícone
	UPFC	14	1	-	
	VSConverter	15	1	-	
Controls	CapControl	16	0	I, S, F,V	
	ExpControl	17	0	-	
	Fuse	18	0	I, S, F,V	
	GenDispatcher	19	0	-	
	InvControl	20	0	-	
	Recloser	21	0	I, S, F,V	
	RegControl	22	0	I, S, F,V	
	Relay	23	0	I, S, F,V	
	StorageController	24	0	-	
	SwtControl	25	0	I, S, F,V	
	UPFCControl	26	0	-	

Tipo	Nome elemento	Id	Numero de conexões	Funções	Ícone
Meters	EnergyMeter	27	0	I, S, F,V	
	Monitor	28	0	I, S, F,V	
	Sensor	29	0	I, S, F,V	
General	CNData	30	0	-	
	GrowthShape	31	0	-	
	LineCode	32	0	v	
	LineGeometry	33	0	-	
	LineSpacing	34	0	-	
	LoadShape	35	0	I, S, F,V	
	PriceShape	36	0	-	
	Spectrum	37	0	-	
	TCC_Curve	38	0	-	
	TSDData	39	0	-	

Tipo	Nome elemento	Id	Numero de conexões	Funções	Ícone
	TShape	40	0	-	
	WireData	41	0	-	
	XfmrCode	42	0	-	
	XYCurve	43	0	I, S, F,V	

Fonte: Produção do próprio autor

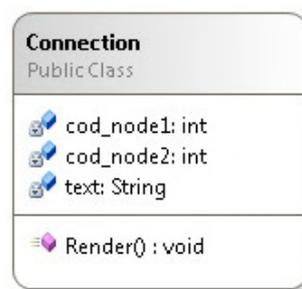
Adicionalmente aos elementos de OpenDSSDirect, programou-se a classe *BusBar* que representa o tipo de elemento barra de conexão, e que não está explícita no modelo OpenDSS. Na Tabela 7 nota-se que existem elementos com 0, 1 e 2 conexões, no caso particular do transformador (*Transformer*) que pode ter mais de um secundário, o programa limitará a funcionalidade a duas conexões, um primário e um secundário. As funções indicadas na Tabela 7 referem-se a funções I, que retornam valores inteiros do tipo *integer*; funções S, que retornam cadeias de caracteres como palavras; funções F, que retornam valores reais do tipo *float*; e funções do tipo V, que retornam listas de valores, sejam números ou textos. Estas funções recebem um parâmetro de acordo com o tipo de informação requerida. Por exemplo: *BUSS(int n, string arg)*, retorna o nome do enésimo bus. Cada classe de elemento dispõe de funções particulares que estão documentadas na guia de desenvolvimento de OpenDSSDirect²⁴.

²⁴ Disponível no repositório de Sourceforge <https://sourceforge.net/projects/electricdss/>, acessado em 12 fev. 2019

3.3.2 O elemento *Connection*

O elemento *Connection* corresponde a uma classe gerada no programa DiagramaBarra para representar a ligação entre elementos do sistema a ser montado. O modelo apresenta-se na Figura 4. Esta classe somente armazena o índice dos elementos que estão ligados e o método para desenhar a linha de conexão na tela.

Figura 4 – Classe *Connection*

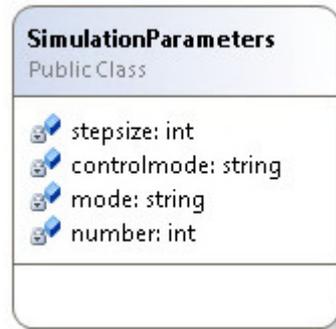


Fonte: Produção do próprio autor

3.3.3 O elemento *SimulationParameters*

A biblioteca OpenDSSDirect possui um total de 110 parâmetros para controlar a simulação. Considerando o caráter pedagógico do *software*, no programa DiagramaBarra somente foram adotados 4 parâmetros do total, sendo implementado na classe *SimulationParameters* (Ver Figura 5). Os parâmetros escolhidos são:

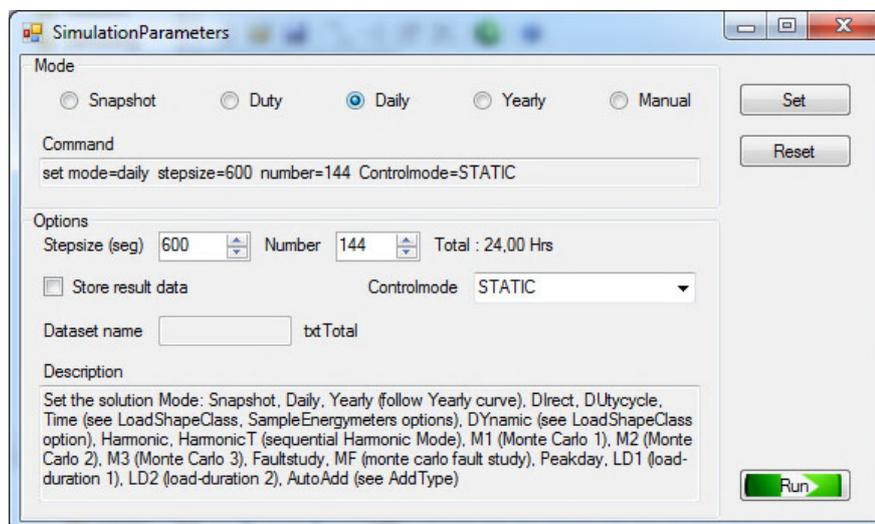
- a) Número de amostras = *number* ;
- b) Intervalo de amostras = *stepsize* ;
- c) Modo de simulação = *mode* ;
- d) Modo de controle = *controlmode* .

Figura 5 – Classe *SimulationParameters*

Fonte: Produção do próprio autor

Estes parâmetros permitem configurar sequencias de ciclos de simulação e não são necessários para o MODO BÁSICO da interface. O painel de opções mostrado na Figura 6 orienta o usuário no significado dos parâmetros, facilitando a compreensão na leitura dos resultados.

Figura 6 – Painel de parâmetros de simulação



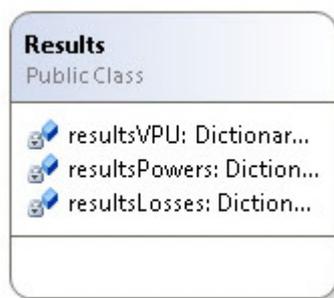
Fonte: Produção do próprio autor

3.3.4 O elemento *Results*

A classe *Results* tem como finalidade armazenar os resultados da simulação. O diagrama mostra-se na Figura 7. A biblioteca OpenDSSDirect fornece todos os valores de corrente e tensão da solução das equações geradas a partir do

diagrama. Sendo o caráter pedagógico do programa DiagramaBarra, se registram o conjunto de valores de magnitude e fase das tensões de cada barra na variável *resultsVPU*, o conjunto de valores das potências ativa e reativa dos elementos de transmissão na variável *resultsLosses*, e o resto de potências ativas e reativas na variável *resultsPowers*. Caso exista carga desbalanceada, serão mostrados os resultados de cada fase. No caso de cargas balanceadas, será mostrado o resultado trifásico total.

Figura 7 – Classe *Results*



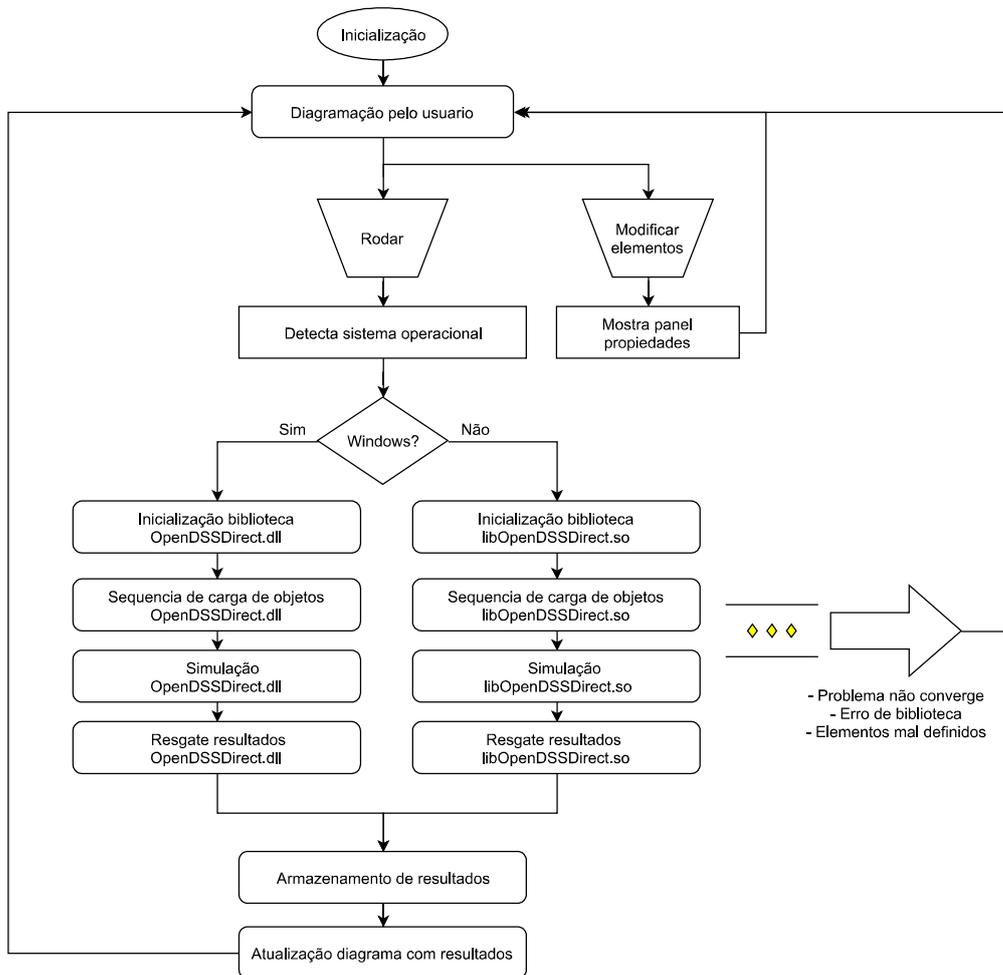
Fonte: Produção do próprio autor

A apresentação dos resultados realiza-se no mesmo diagrama, facilitando a leitura e compreensão.

3.4 DESENVOLVIMENTO DA SEQUÊNCIA DE SIMULAÇÃO

No contexto de um novo programa, a curva de aprendizado de um usuário depende da facilidade de utilizar os recursos disponíveis para uma finalidade particular. Com o intuito de simplificar os passos do usuário do programa DiagramaBarra, esquematiza-se as diferentes funções que procedem na simulação de redes de energia: abordagem do problema, elaboração da topologia no *software*, definição de parâmetros, execução e recopilação de resultados. Usando um esquema similar, propõe-se a sequência ilustrada na Figura 8 para ser utilizada no desenvolvimento do programa DiagramaBarra.

Figura 8 – Fluxograma de montagem e simulação



Fonte: Produção do próprio autor

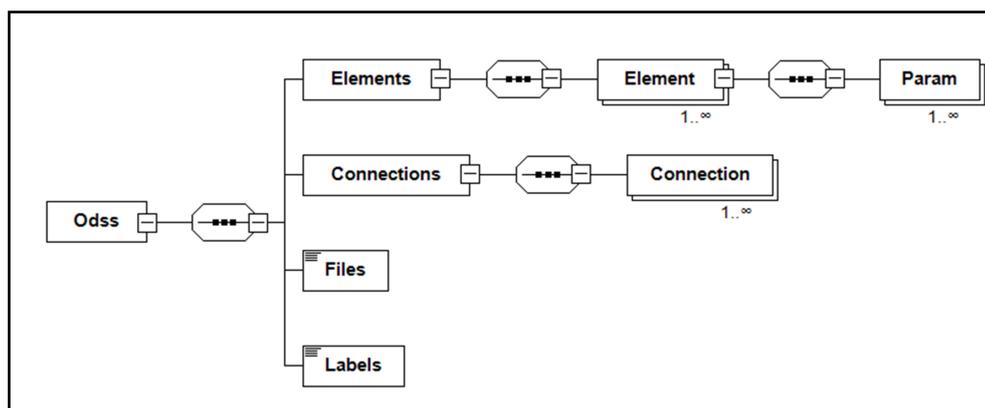
Pode notar-se que a função do usuário limita-se a diagramar e definir parâmetros dos elementos, reduzindo a complexidade no uso da ferramenta de *software*. Neste processo de simplificação, elimina-se a necessidade de aprender uma nova linguagem ou sintaxe como sucede no OpenDSS ou GridLAB-D (Ver Tabela 3). Este esquema complementa-se com a ajuda contextual presente na interface para a conexão dos elementos e configuração de parâmetros.

3.5 DESENVOLVIMENTO DO MODELO DE ARQUIVOS

Adota-se o formato de arquivos com esquema XML mostrado na Figura 9 com as recomendações da IEEE para ODM (*Open Model for Exchanging Power*

Simulation Data), um esquema que ainda está em desenvolvimento (MILANO, ZHOU e HOU, 2009). O programa DiagramaBarra utiliza uma versão simplificada do formato ODM para conter os elementos com seus parâmetros, as conexões lógicas, dados anexos codificados e uma expansão para incluir textos dinâmicos no futuro.

Figura 9 – Esquema XML



Fonte: Produção do próprio autor

Os arquivos de dados anexos como curvas de carga no tempo, ou curvas de irradiância solar ou curvas de vento ficam armazenados no formato XML de forma comprimida, permitindo a portabilidade do sistema de forma íntegra.

3.6 DESENVOLVIMENTO DO MODELO DE EXECUÇÃO

OpenDSS não fornece uma versão para Linux ou MacOS, e DiagramaBarra se concebe dentro do conceito multiplataforma, portanto para poder incluir a biblioteca no programa foi necessário adaptar-se.

O programa DiagramaBarra usa a biblioteca dinâmica OpenDSSDirect como recurso para realizar a simulação e cálculo dos resultados. EPRI disponibiliza essa biblioteca no pacote de instalação do OpenDSS e entrega vários exemplos em diversas linguagens de programação. A documentação cita duas modalidades de vinculação da biblioteca OpenDSS. Uma mediante chamadas COM (*Component Object Model*), tecnologia de *software* usada nos produtos Microsoft que provê acesso aos objetos da biblioteca, suas propriedades e funções, porém requer permissão administrativa para instalação e apresenta conflito com chamadas assíncronas. Outro modo de vinculação é mediante chamadas diretas a biblioteca

dinâmica DLL, que requer a declaração formal de todas as funções e objetos, sendo esta a modalidade utilizada neste projeto.

Através dos códigos fontes disponíveis, usuários da comunidade disponibilizaram um procedimento para compilar a biblioteca *OpenDSSDirect* no Linux 64bit e no MacOS 64bit. Fazendo possível gerar a biblioteca *libOpenDSSDirect.so* usando o compilador Free Pascal Compiler²⁵. Esta biblioteca foi incluída de forma binária no pacote de instalação do DiagramaBarra,

Assim foi possível desenvolver o programa DiagramaBarra com a capacidade de distinguir o sistema operacional na qual se está executando e assim, carregar a biblioteca da plataforma correspondente no tempo de execução. Tendo desta maneira a capacidade de instalar-se indistintamente em qualquer dos sistemas operacionais suportados.

Adicionalmente, foram implementadas funções extras dentro da biblioteca *OpenDSSDirect.so* para dispor de resultados que não estavam acessíveis por incompatibilidades na gestão de memória de Mono dentro do sistema operacional Linux²⁶.

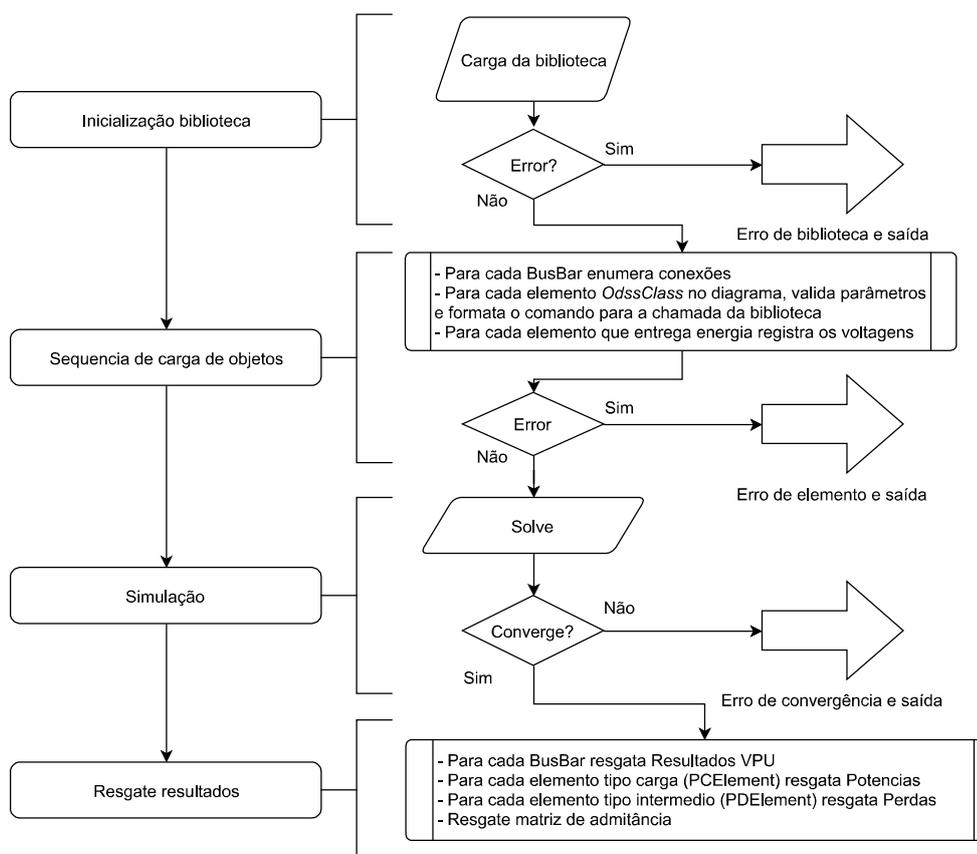
As funções extras foram programadas dentro da função de tipo F da classe *BusBar* $BUSF(int\ n, float\ f)$, mediante uma expansão do parâmetro n , permitindo recuperar corretamente magnitudes de tensão e ângulo dos resultados. A mesma modificação foi realizada na função de tipo F dos elementos de transmissão $PCElementF(int\ n, float\ f)$ permitindo a resgatar os valores de potência e reativa da simulação.

No processo de execução inicializa-se a comunicação com a biblioteca *OpenDSSDirect* e no caso de erro, aborta-se a sequência. Logo, gera-se o comando de chamada para cada elemento do diagrama, de forma ordenada de acordo com o tipo de elemento. Durante esta etapa, se existe um erro de chamada, aborta-se a sequência. No passo seguinte, executa-se o comando para resolver o circuito. No caso de que o problema não possuir convergência, se detém a sequência. Finalmente, os resultados de tensão, potências, perdas e a matriz de admitância registram-se na memória. O modelo de execução segue o fluxograma mostrado na Figura 10.

²⁵ Procedimento disponível em <https://github.com/Muxelmann/OpenDSSDirect.make>

²⁶ Mono ainda esta em desenvolvimento.

Figura 10 – Modelo de execução simulação



Fonte: Produção do próprio autor

A interface geral realiza uma validação de cada passo do usuário. Para cada caso de falha no processo, apresenta-se um quadro de mensagem de advertência com o motivo do cancelamento da simulação.

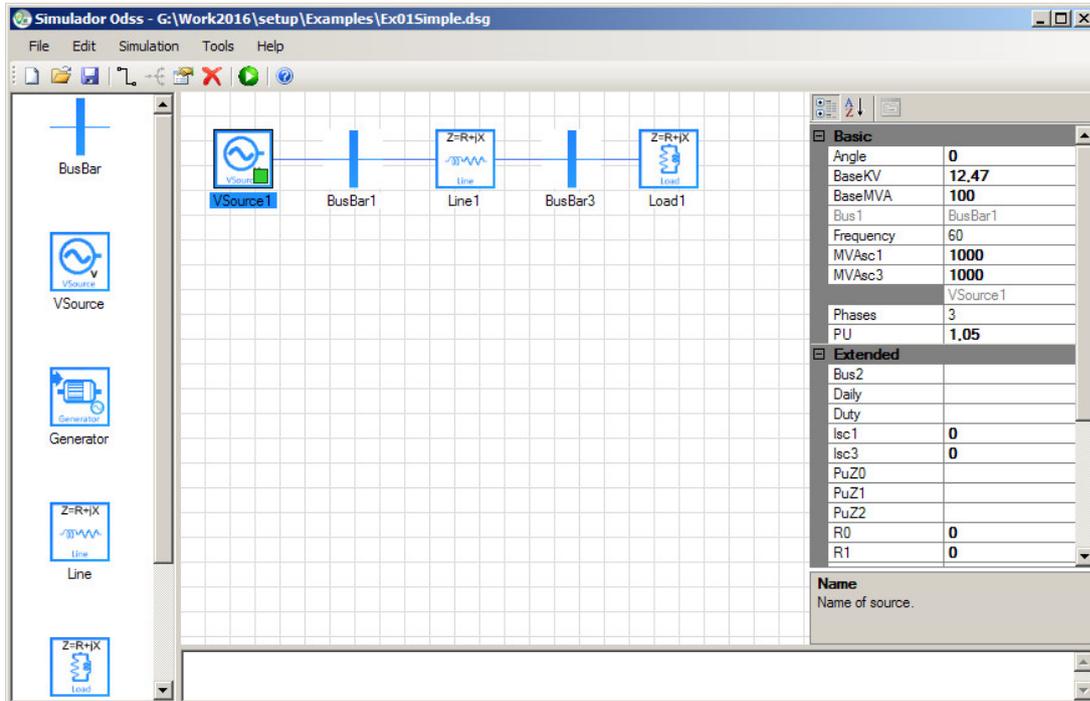
3.7 INTERFACE

A partir dos esquemas previamente descritos gerou-se uma interface gráfica para conter visualmente a informação de topologias, resultados e parâmetros de maneira acessível. A interface apresentada na Figura 11, consta de área de trabalho, menu de usuário, barra de ferramentas, barra de componentes, painel de propriedades, e um quadro de mensagens.

O menu de usuário permite acessar a funções de arquivos, modificar elementos, simular e ferramentas de apoio. O menu programou-se com recursos de internacionalização permitindo mudar para os idiomas inglês, português e espanhol,

sendo possível incorporar mais idiomas, procedimento que pode ser realizado pelo usuário sem requerer compilação. O formato dos arquivos de idiomas está descrito no Anexo B.

Figura 11 – Interface principal



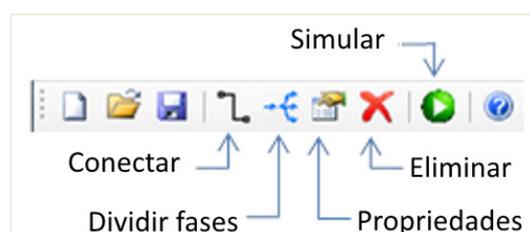
Fonte: Produção do próprio autor.

Utilizaram-se características de programação que estão presentes no *Simulink* e *PSim*. Mediante ações de *drag'n drop* o usuário arrastra ícones da barra de componentes para a área de trabalho. Cada elemento possui uma etiqueta identificadora que permite modificar o nome. O usuário pode colocar o componente em qualquer posição da área de trabalho. Mediante seleção do elemento, habilita-se a função para conectar com outro elemento. O DiagramaBarra somente permite conexões de acordo com a Tabela 7, evitando erros do usuário. Os parâmetros de cada elemento são acessíveis mediante a ação do duplo click ou pelo botão da barra de ferramentas. O painel de propriedades aparece na seção à direita como se vê na Figura 11. Cada propriedade apresenta uma explicação que se mostra no quadro descritivo.

3.7.1 Barra de Ferramentas

A Figura 12 mostra a Barra de Ferramenta que permite a conexão entre elementos, mostrar o painel de propriedades, eliminar um elemento ou conexão selecionada. Também contém um botão correspondente à ação de executar a simulação do diagrama.

Figura 12 – Barra de ferramentas



Fonte: Produção do próprio autor.

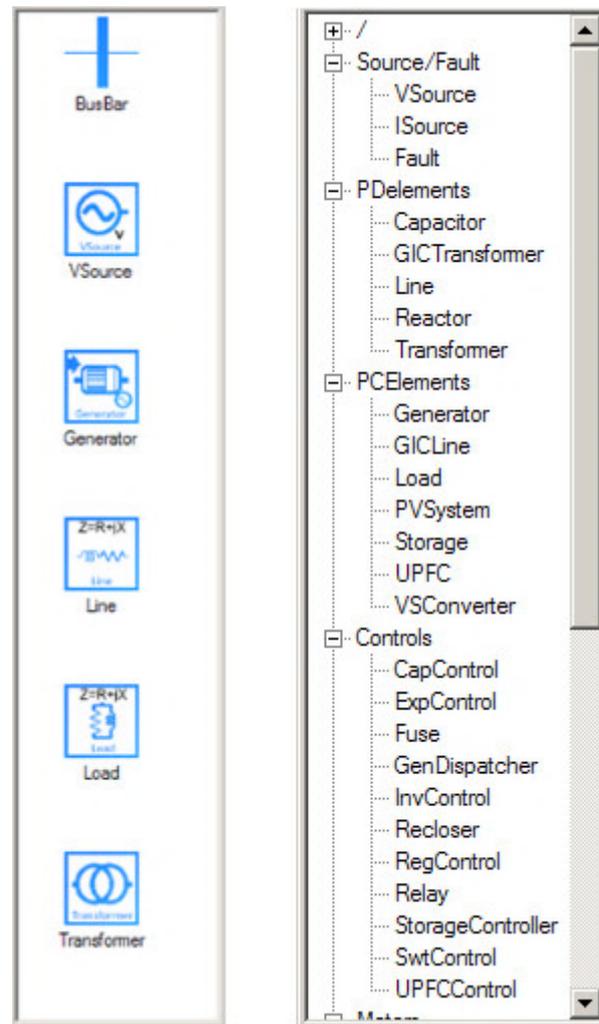
Os botões tem comportamento dinâmico conforme seja selecionado um elemento ou conexão do diagrama. No caso de ser selecionado um elemento do tipo *Load* (Id 11 da Tabela 7) que corresponde a uma carga de consumo, habilita-se o botão *Dividir fases*, que realiza a decomposição de uma carga trifásica equilibrada em três cargas monofásicas desequilibradas. A ação efetua as conexões das respectivas cargas para após isto, realizar os cálculos de fluxo de potência e de tensões para cada uma das três fases.

3.7.2 Barra de Componentes

A Barra de Componentes exhibe os tipos de elementos disponíveis para realizar um diagrama. Apresenta-se de duas formas, no a) Modo Básico, e no b) Modo Avançado como se ilustra na Figura 13. Este último modo é ativado mediante o Menu Ferramentas (ou *Tools*).

No Modo Básico, a Barra de Componentes contém os elementos mais comuns usados nos exercícios de sistemas de energia. Neste modo se dispõe da barra (*BusBar*), fonte de barra alimentadora (*VSource*), geradores (*Generator*), linhas de transmissão (*Line*), transformadores (*Transformer*) e cargas (*Load*).

Figura 13 – Barra de componentes



a) Modo Básico

b) Modo Avançado

Fonte: Produção do próprio autor.

Sendo que um dos objetivos contempla a possibilidade do uso do software pelo profissional formado, considerou-se incluir o acesso a todos os elementos disponíveis da biblioteca para simulação no Modo Avançado. Usa-se a mesma interface, porém os elementos da Barra de componentes apresentam-se classificados por tipo e organizados na forma de uma árvore.

3.7.3 Painel de Propriedades

A Figura 14 mostra o painel de propriedades para o elemento *Line*. Na programação orientada a objetos, cada classe contém propriedades e cada propriedade possui atributos. Usando esta característica, os parâmetros dos

elementos e sua descrição ficam disponíveis ao usuário mediante a seleção do elemento no diagrama.

Figura 14 – Painel de propriedades

Basic	
Bus1	BusBar1
Bus2	BusBar3
C0	1,6
C1	3,4
Length	5,28
Linecode	
	Line1
Phases	3
R0	0,17
R1	0,06
Units	kft
X0	0,4
X1	0,12
Extended	
B0	0
B1	0
Cmatrix	
Cncables	
EarthModel	
Geometry	
Rg	0
Rho	0
Rmatrix	
Spacing	
Switch	False
Tscables	
Wires	
Xg	0
Xmatrix	
Name Name of Line.	

Fonte: Produção do próprio autor.

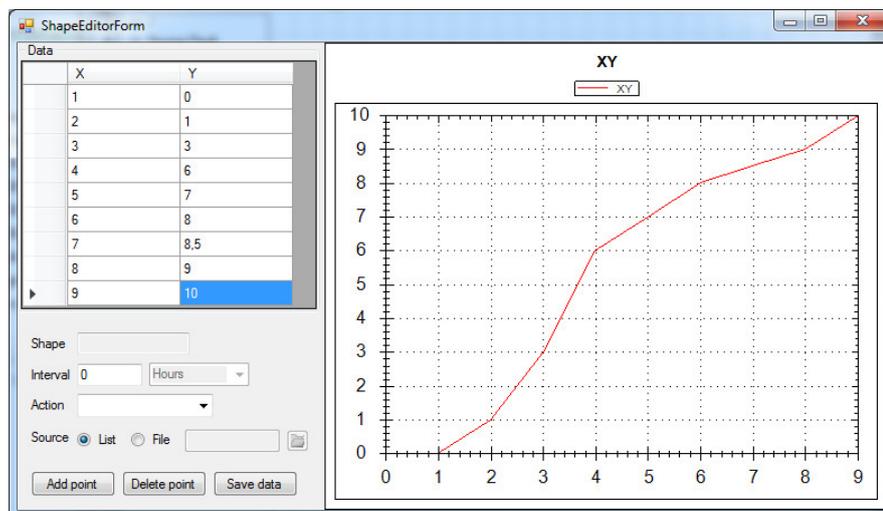
3.8 FERRAMENTAS ADICIONAIS DESENVOLVIDAS

Anexos à interface principal, foram desenvolvidas outras ferramentas de apoio para o usuário com o propósito de facilitar a integração dos elementos. Estas estão acessíveis através do menu *Ferramentas*.

3.8.1 Editor de comportamento variável

Na solução de sistemas de energia, alguns elementos podem ter parâmetros variáveis no tempo, como o consumo de uma cidade no transcurso de um dia. O programa DiagramaBarra entrega ao usuário a possibilidade de variar parâmetros de forma automática usando o elemento *LoadShape* (elemento Id 35 da Tabela 7) disponível na Barra de componentes no Modo avançado. O editor de curvas mostrado na Figura 15 permite gerar de forma visual os parâmetros usados no elemento. Dados podem ser importados desde arquivos externos, sendo incorporados no diagrama. Este painel utiliza a biblioteca *ZedGraph* para plotar os dados. A ativação do elemento *LoadShape* na simulação funciona em conjunto com o modo de simulação *Daily*, *Yearly* e *Duty* do painel de parâmetros de simulação (Ver Figura 5)

Figura 15 – Editor de curvas



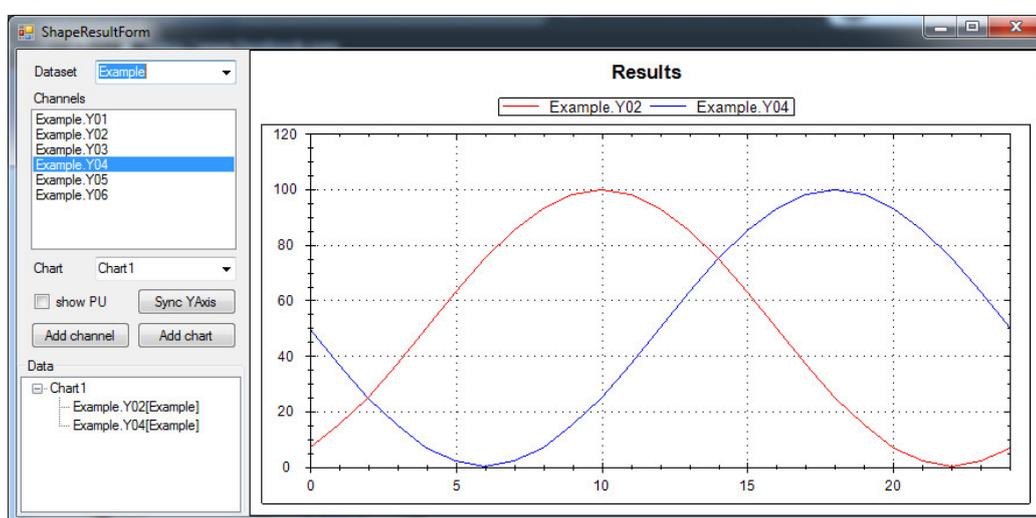
Fonte: Produção do próprio autor.

3.8.2 Painel de gráficos de resultados

A simulação utilizando o elemento *LoadShape* citado previamente, entrega resultados na forma de séries de valores, que requiere incluir o elemento *Monitor* no diagrama (Id 28 da Tabela 7). Para visualizar as variáveis registradas no *Monitor* gerou-se o painel de gráfico de resultados mostrado na Figura 16. A apresentação

dos resultados na interface gráfica somente considera alguns valores do conjunto disponível como foi citado na Seção 3.3.4. Porém, o elemento *Monitor* possui uma estrutura de registro independente na forma de listas de valores indexados. A partir dessa propriedade, sequências de valores obtidos na simulação ficam acessíveis para plotar ou exportar. Adicionalmente, a ferramenta de gráfico de resultados registra diferentes iterações de simulação que o usuário realiza, possibilitando comparar curvas de simulações distintas que foram armazenadas previamente. Por exemplo, poder-se-ia comparar as tensões em uma determinada barra para dois ou mais casos.

Figura 16 – Gráfico de resultados



Fonte: Produção do próprio autor.

3.8.3 Painel da matriz de admitância

Na solução de problemas de sistemas de energia, podem ser utilizados métodos da álgebra matricial para a solução sistemas lineares. A construção da matriz de admitâncias corresponde a um dos objetivos de uma das disciplinas do curso de graduação de engenharia elétrica da UDESC. O Painel mostra-se na Figura 17.

Esta ferramenta constitui um apoio didático aos conteúdos ministrados, permitindo verificar a formação de tal matriz, que é base para o cálculo de fluxo de potência. Inclui também uma utilidade adicional que possibilita inverter a matriz de admitâncias para obter a matriz de impedâncias do sistema. A matriz de

impedâncias, juntamente com a matriz de admitâncias, é muito utilizada para o cálculo de correntes de curto-circuito (GRAINGER e STEVENSON, 1994).

Figura 17 – Painel da matriz de admitância

	BUSBAR1.1	BUSBAR2.1	BUSBAR3.1	BUSBAR4.1	BUSBAR5.1	BUSBAR6.1	BUSBAR7.1
BUSBAR1.1	$21.010 + j(-76.276)$	$-0.385 + j(1.923)$	-	-	-	-	-
BUSBAR2.1	$-0.385 + j(1.923)$	$0.769 + j(-3.846)$	$-0.385 + j(1.923)$	-	-	-	-
BUSBAR3.1	-	$-0.385 + j(1.923)$	$0.769 + j(-3.846)$	$-0.385 + j(1.923)$	-	-	-
BUSBAR4.1	-	-	$-0.385 + j(1.923)$	$0.769 + j(-3.846)$	$-0.385 + j(1.923)$	-	-
BUSBAR5.1	-	-	-	$-0.385 + j(1.923)$	$0.769 + j(-3.846)$	$-0.385 + j(1.923)$	-
BUSBAR6.1	-	-	-	-	$-0.385 + j(1.923)$	$0.769 + j(-3.846)$	$-0.385 + j(1.923)$
BUSBAR7.1	-	-	-	-	-	$-0.385 + j(1.923)$	$0.404 + j(-1.923)$

Fonte: Produção do próprio autor.

3.8.4 Editor de grupos

O Editor de Grupos mostrado na Figura 18 é uma utilidade incorporada para facilitar a edição de diagramas. Na maioria das referências de sistemas elétricos de potência, a topologia do sistema, assim como seus dados de linha de transmissão, cargas, transformadores e geradores são apresentados na forma de tabela de valores. Portanto concebe-se este editor que dispõe um acesso aos parâmetros de todos os elementos do mesmo diagrama agrupados por tipo. A modificação dos parâmetros no Editor de grupos reflete-se no diagrama e vice-versa.

Adicionalmente os dados podem ser exportados e importados de arquivos de planilha Excel no formato CSV.

Figura 18 – Editor de listas

Name	Bus1	Bus2	Linecode	Length	Phases	R1	X1	R0	X0	C1	C0
Line1	BusBar4	BusBar5		100	3	0.0529	0.4494	0.13225	0.8972	8.82	5.188
Line2	BusBar4	BusBar6		100	3	0.08993	0.4863	0.224825	1.2139	7.922	4.74
Line3	BusBar5	BusBar7		100	3	0.16928	0.8516	0.4232	2.1262	15.34	9.025
Line4	BusBar6	BusBar9		100	3	0.20631	0.8972	0.5157	2.2959	17.95	10.55
Line5	BusBar8	BusBar9		100	3	0.062951	0.5331	0.15737	1.3307	10.47	6.15
Line6	BusBar7	BusBar8		100	3	0.044965	0.3808	0.11241	0.7615	7.471	4.394

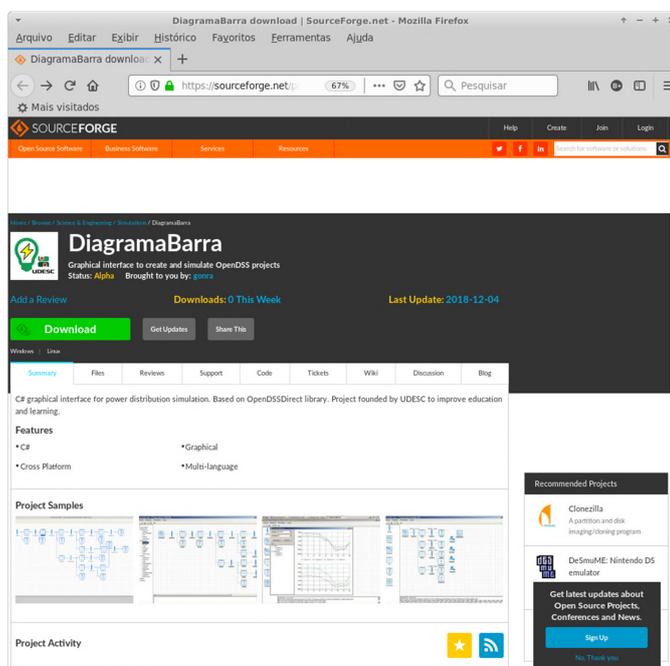
Fonte: Produção do próprio autor.

3.9 DISTRIBUIÇÃO E DOCUMENTAÇÃO

Dentro do conceito de *software* livre, está presente o acesso à informação e a disponibilização dos códigos fontes. Para este propósito existem vários sítios Web orientados a compartilhar e difundir novos desenvolvimentos e que incorporam ferramentas para controle de descargas, estatísticas e utilidades para publicar documentação.

O programa DiagramaBarra está disponível para a descarga na forma de instaladores e como arquivos comprimidos no site SourceForge (Ver Figura 19).

Figura 19 – Webpage do DiagramaBarra no SourceForge



Fonte: www.sourceforge.net/projects/diagramabarra

Note-se que nesta plataforma também estão disponíveis para descarga os códigos fonte de OpenDSS.

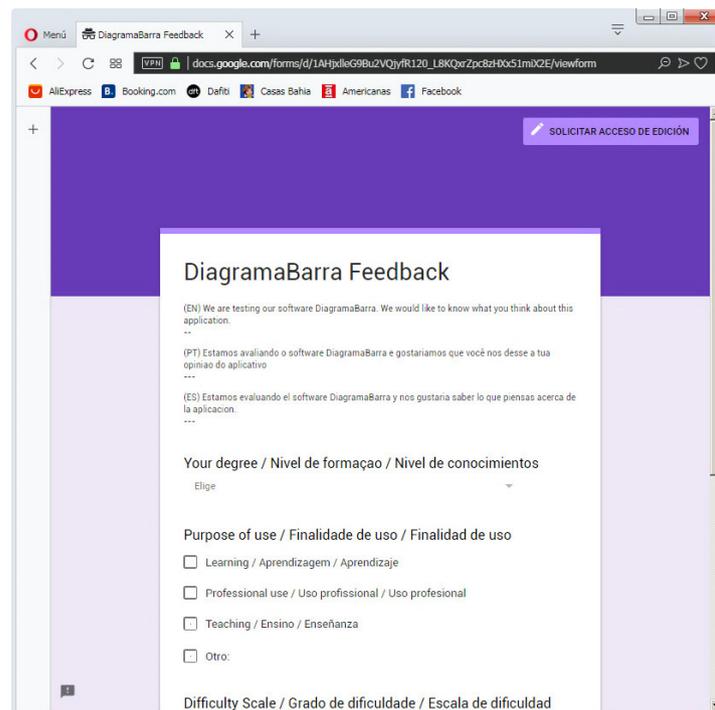
A documentação realizada se publicou no formato *Wiki*, o qual permite inclusão de recursos gráficos, multimídia e vínculos para outras plataformas de difusão.

Para a avaliação do programa DiagramaBarra considerou-se uma simples enquete de percepção do usuário que se abre na primeira vez que o programa se executa, ficando também acessível através do menu de usuário *Ajuda*→*Feedback*.

O questionário se realiza de forma anônima na plataforma gratuita Google Forms²⁷ (Ver Figura 20). O seguimento dos resultados permitirá a futuro realizar melhorias e correções na interface.

Finalmente, o usuário pode verificar a disponibilidade de atualizações do programa DiagramaBarra mediante o menu de usuário *Ajuda*→*Atualizações*.

Figura 20 – Enquete de percepção do usuario

The image shows a browser window displaying a Google Forms survey. The browser's address bar shows the URL: docs.google.com/forms/d/1AHjXlleG9Bu2VQjy/R120_L8KQwZpc8zH0xS1miXCE/viewform. The survey title is "DiagramaBarra Feedback". It includes three introductory paragraphs in English, Portuguese, and Spanish. Below the text, there are three sections: "Your degree / Nivel de formação / Nivel de conocimientos" with a dropdown menu labeled "Elige"; "Purpose of use / Finalidade de uso / Finalidad de uso" with four checkboxes: "Learning / Aprendizagem / Aprendizaje", "Professional use / Uso profissional / Uso profesional", "Teaching / Ensino / Enseñanza", and "Otro"; and "Difficulty Scale / Grado de dificultad / Escala de dificultad". A purple button labeled "SOLICITAR ACCESO DE EDICIÓN" is visible in the top right corner of the form area.

Fonte: Google Forms

²⁷ Disponível em <http://forms.google.com>

4 USO DO PROGRAMA DIAGRAMABARRA

4.1 DIFERENÇAS E MELHORIAS

A arquitetura aberta e modular do programa DiagramaBarra usa a biblioteca OpenDSSDirect para o cálculo matemático, da mesma forma que o OpenDSS, porém existem diferenças na estrutura de funcionamento.

OpenDSS surgiu a partir do *software* DSS que estava programado em C++. Quando foi adquirido pela EPRI, migrou-se para a linguagem de programação orientada a objetos Delphi. Isso permitiu um programa veloz, leve e compacto, que pode executar em qualquer computador com Microsoft Windows. Posteriormente passa a domínio público com a publicação dos códigos fontes, a partir da qual recebe contribuições na programação de novas funções. Recentemente disponibilizou-se uma nova versão chamada OpenDSS-G baseada na interface LabView²⁸, *software* comercial que tem maiores requerimentos de *hardware* e *software*.

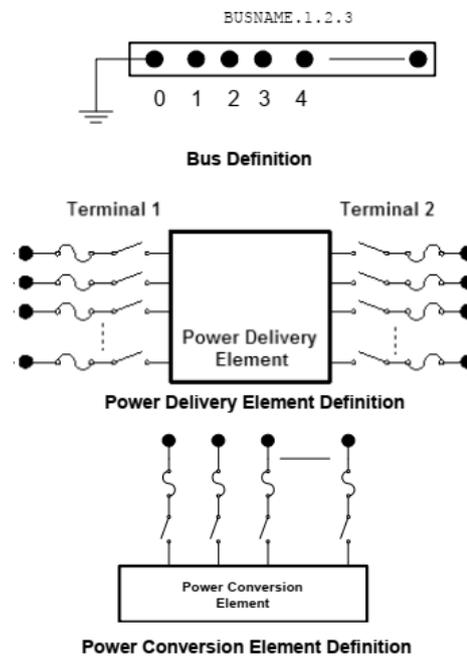
Na documentação disponibilizada por EPRI (2012), descreve-se OpenDSS como uma ferramenta de simulação de sistemas elétricos para sistemas de distribuição. Providencia uma interface de usuário como um editor de textos. O programa interpreta comandos ingressados pela interface ou mediante a carga de *scripts* de comandos e constrói na memória a forma matricial do sistema de equações do circuito. A solução do sistema se consegue usando a biblioteca KLU²⁹ ficando os resultados acessíveis através de menus, os quais geram arquivos de texto.

Prévio ao uso do *software*, o usuário deve entender a sintaxe dos comandos e conceitos básicos da definição de *Buses*, *PDElements* (elementos de entrega de energia) e *PCElements* (elementos de conversão de energia) como está ilustrado na Figura 21. A documentação básica para treinamento é extensa, e representa uma pequena porção das possibilidades de uso do *software*.

²⁸ LabView de National Instruments, disponível em <http://www.ni.com>

²⁹ Biblioteca da Universidad de Florida. Disponível como software livre <https://sourceforge.net/projects/klusolve/>

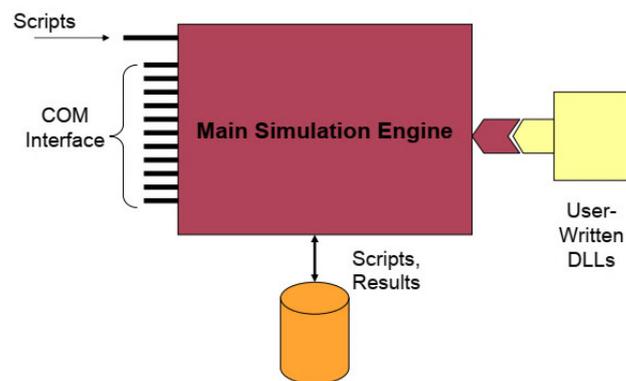
Figura 21 – Definições OpenDSS



Fonte: Manual ODSS

A estrutura do programa OpenDSS ilustrada na Figura 21 incorpora o acesso a funções descrito previamente na Seção 3.2.

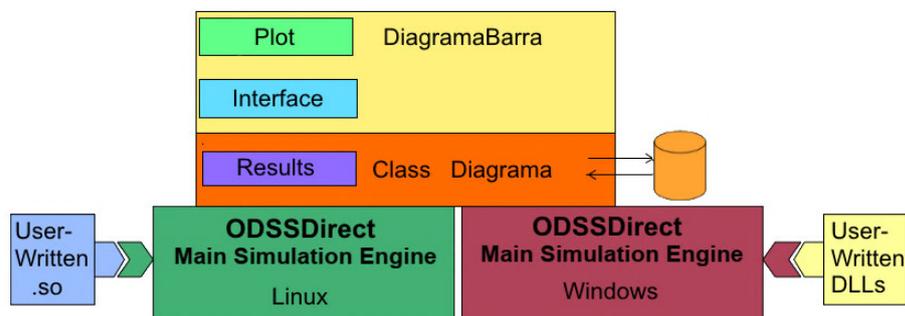
Figura 22 – Estrutura OpenDSS



Fonte: OpenDSS Level 1 Training

Por outro lado a estrutura do programa DiagramaBarra mostrado na Figura 22, simplifica o esquema do OpenDSS. A interface gráfica permite prescindir do uso de formatação e sintaxe. No programa DiagramaBarra os parâmetros dos elementos contam com ajuda contextual e estão sempre disponíveis através painel de propriedades. Com a programação multiplataforma, desenvolvida neste trabalho o DiagramaBarra expande o uso do OpenDSS a outro sistema operacional.

Figura 23 – Estrutura do programa DiagramaBarra

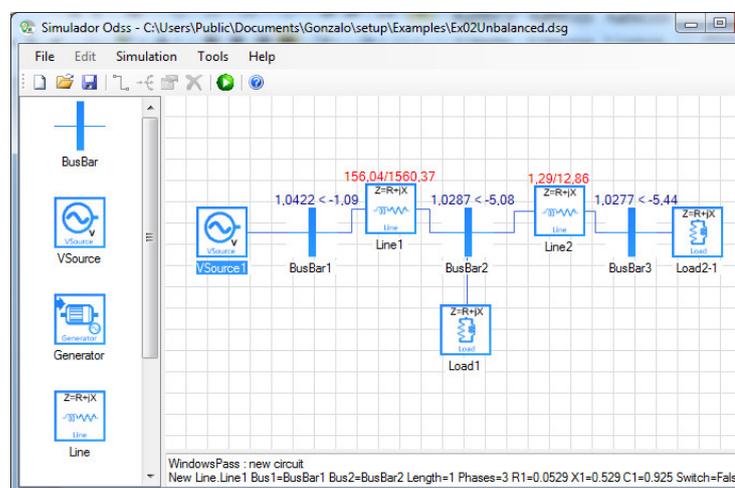


Fonte: Produção do próprio autor

Devido à utilização da plataforma .NET Framework para programar o DiagramaBarra, os componentes gerados passam a formar uma biblioteca de categoria .NET, permitindo a sua utilização para desenvolver outros aplicativos através da chamada a objetos por referência do Visual Studio.

A entrega de resultados na mesma interface do programa DiagramaBarra, apresenta vantagens e desvantagens respeito do OpenDSS. A principal vantagem é a rápida visualização do elemento de interesse como se aprecia na Figura 24, onde se mostram as tensões de barra e as perdas de potencia ativa e reativa nos elementos que formam a rede elétrica, neste caso, nos segmentos de linha de transmissão. Por outro lado, como OpenDSS carece de uma interface de diagramação, a geração de resultados se realiza através de muitos arquivos, um arquivo para cada tipo de análise, muitas vezes confundindo o usuário.

Figura 24 – Resultados no DiagramaBarra



Fonte: Produção do próprio autor

4.2 APLICAÇÃO NO ENSINO

O propósito do programa DiagramaBarra é servir como apoio na aprendizagem. De acordo com os planos de ensino do Departamento de Engenharia Elétrica (DEE) da UDESC, identificam-se as seguintes disciplinas que têm conteúdos relacionados com sistemas de potência.

Tabela 8 – Disciplinas com conteúdos afins com a aplicação do DiagramaBarra e suas ementas

Disciplina	Ementa
Geração de energia elétrica (ET1GEE1)	Panorama das principais fontes de energia elétrica. Sistemas centralizados e descentralizados de geração de energia elétrica. Geração hidroelétrica. Geração termelétrica. Sistemas não convencionais de geração elétrica. Energia solar e Fotovoltaica. Energia eólica e aero geradores. Perspectivas e tendências da geração de eletricidade.
Proteção de sistemas elétricos (ET3PSE1)	Representação de sistemas elétricos. Componentes simétricos. Cálculo de curto-circuito. Transformadores para instrumentos. Proteção de sistemas elétricos de potência. Relés. Coordenação da proteção.
Sistemas de energia elétrica (ET2SEE2)	Modelagem de barras, linhas, transformadores em cargas. Considerações operacionais sobre o sistema de energia. O problema de cálculo do fluxo de potência. Compensação reativa. Transformadores reguladores e defasadores. Equações estáticas de fluxo de potência (EEFP). Métodos iterativos de para a solução das EEFP. Fluxo de potência CC. Tópicos em estabilidade em sistema de potência.

Tabela 8 – Disciplinas com conteúdos afins com a aplicação do software e suas ementas (continuação)

Disciplina	Ementa
Eficiência energética (ET3EFE1)	Energia e Sociedade; Fontes de Energia Convencionais, Fontes de Energia não convencionais, Aspectos econômicos, Eficiência energética; Tecnologias de Racionalização de Energia Elétrica.
Transmissão e distribuição de energia (TDE0002)	Organização do Setor Elétrico Brasileiro. Características de um sistema de distribuição. Subestações. Indicadores de qualidade da energia elétrica. Normas de fornecimento de energia elétrica. Estrutura tarifária. Aspectos construtivos e modelagem das linhas de transmissão de energia elétrica. Operação e desempenho das linhas de transmissão. Modelagem dos transformadores de potência. Análise dos sistemas de energia elétrica em p.u. Aspectos básicos da transmissão de energia elétrica em CC.

Disciplina	Ementa
Fonte: UDESC	

Dentre algumas funcionalidades que o programa DiagramaBarra poderá contribuir para as disciplinas supracitadas, considerando apenas os tópicos atualmente pertencentes às mesmas, estão:

- a) Na disciplina de Geração de Energia Elétrica: Após dimensionar sistemas de geração fotovoltaicos e de energia eólica, a implementação dos mesmos no DiagramaBarra ajudará na compreensão de efeitos que estas fontes de energia, que possuem variabilidade, podem ocasionar nas tensões do sistema a que estão conectadas.
- b) Na disciplina de Sistemas de Energia Elétrica: Aplicação de modelagens de diversos dispositivos tais como linhas de transmissão, transformadores, fontes de geração de energia convencionais e não convencionais, controle de potência e tensão, geradores, fluxo de potência em redes balanceadas e desbalanceadas mono ou polifásicas, impactos da inserção de novas fontes de energia no sistema elétrico, inclusive do sistema de carregamento de veículos elétricos, e de dispositivos do tipo *Flexible Alternating Current Transmission Systems* (FACTS). Alocação ótima de geradores e compensadores estáticos e novas fontes de energia no sistema. Estudos básicos de estabilidade transitória eletromecânica.
- c) Na disciplina de Transmissão e Distribuição de Energia Elétrica: Aplicação de diversas configurações geométricas cabos na modelagem de linhas de transmissão e transformadores. Cálculo de indicadores de qualidade de energia e de confiabilidade via emulação de descontinuidade do serviço, podendo ainda considerar os custos de interrupções.
- d) Na disciplina de Proteção de Sistemas Elétricos: Representação de sistemas mono e polifásicos, balanceados ou não. Cálculo de curtos circuitos em sistemas polifásicos balanceados ou não. Cálculo das contribuições para as correntes de curto circuito das novas fontes de energia (fotovoltaica e eólica, com ou sem banco de baterias).

Contribuição para a corrente de curto circuito de um sistema de armazenamento de energia (estações de carregamento de carros elétricos); Impacto da inserção das novas fontes de energia principalmente no ajuste de proteção de sistemas de distribuição onde poderão ocorrer situações de fluxo de potência bidirecional. Coordenação da proteção com e sem a presença de geração distribuída.

Porém, vale destacar que aplicações do Diagrama Barra não se limitam apenas às disciplinas mencionadas. Existem aplicações do OpenDSS [ref] e consequentemente do Diagrama Barra em:

- Planejamento de longo prazo de sistemas com geração distribuída;
- Estimação de estados em sistemas elétricos;
- Estudos de otimização de controle de tensão-potência reativa;
- Armazenamento comunitário de energia;
- Estudos de correntes geomagnéticas induzidas em transformadores e linhas de transmissão;
- Modelagem de sistemas de grande porte considerando simulações em tempo real do tipo *Hardware In the Loop* (HIL) – neste ponto poderiam ser utilizados protótipos de conversores de energia (eletrônica de potência), além de sistemas de acionamento para máquinas elétricas;
 - Automação de sistemas de distribuição;
 - Co-simulação de sistemas de potência e sistemas de informação e comunicações (*Information and Communication Technologies* – ICT);

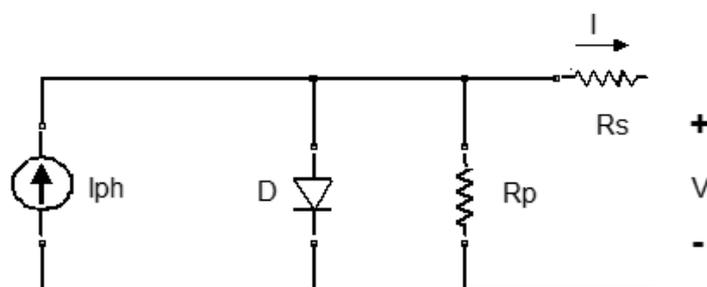
As duas últimas aplicações se agregadas, por exemplo à disciplina de Sistemas Elétricos de Energia, poderiam culminar na criação de uma disciplina de *Smart Grids*, inicialmente oferecida no programa de pós-graduação e em seguida na graduação.

4.3 EXEMPLO DE APLICAÇÃO

Junto ao desenvolvimento do programa DiagramaBarra ocorre o trabalho de análise do impacto de Usina Solar Fotovoltaica (USF) no sistema elétrico. Em vista da variação da irradiância solar durante o dia apresenta-se a oportunidade de efetuar um estudo das perdas presentes em diferentes estratégias de conexão ao sistema de distribuição elétrica.

Inicialmente uma USF consta de conjuntos painéis interligados em forma de rede matricial. Um painel contém múltiplas células ligadas em série e paralelo. Cada célula fotovoltaica do painel entrega uma tensão contínua V_p e uma corrente I_p , dependendo da intensidade solar. O circuito equivalente real de uma célula fotovoltaica pode ser representado como sendo uma fonte de corrente em paralelo com um diodo, uma resistência em série R_S e uma resistência em paralelo R_P conforme ilustrado na Figura 25. Uma célula fotovoltaica possui níveis baixos de tensão (da ordem de 0,7V) e de corrente (da ordem de 3A).

Figura 25 – Circuito equivalente de célula fotovoltaica



Fonte: Produção do próprio autor.

Deste circuito equivalente, através da Lei de Kirchhoff das Correntes, pode-se obter a relação:

$$I = I_{ph} - I_D - I_P \quad (4.1)$$

Onde os parâmetros são:

I_{ph} – corrente por efeito fotoelétrico;

I_D – corrente no diodo D;

I_P – corrente na resistência em paralelo R_P .

As equações (4.2) e (4.3) fornecem as correntes I_D e I_P (ZILLES, MACÊDO, *et al.*, 2012)

$$I_D = I_0 \times \left[e^{\left(\frac{q \times V}{n \times k \times T} \right)} \right] \quad (4.2)$$

$$I_P = \frac{V + (I \times R_S)}{R_P} \quad (4.3)$$

$$I_{ph} = [I_{sc} + \alpha \times (T - T_r)] \times \frac{P}{1000} \quad (4.4)$$

Onde:

I_0 – corrente de saturação reversa da célula;

q – é a carga do elétron igual a $1,6 \times 10^{-19} \text{C}$;

n – o Fator de qualidade da junção p-n;

k – a constante de Boltzmann igual a $1,35 \times 10^{-23}$;

T – a temperatura de trabalho da célula.

I_{sc} – corrente de curto-circuito;

α – coeficiente de temperatura da corrente de curto circuito;

T_r – temperatura de referência da célula igual a 298°K ;

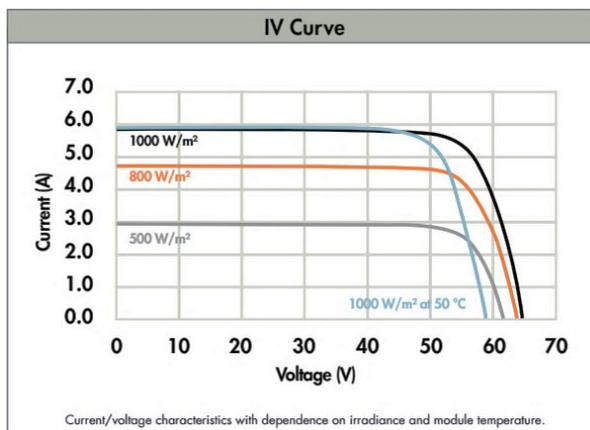
P – irradiância em W/m^2 ;

Substituindo-se as Equações (4.2), (4.3) e (4.4) em (4.1), obtém-se o equacionamento (4.5).

$$I = [I_{sc} + \alpha \times (T - T_r)] \times \frac{P}{1000} - I_0 \cdot \left[e^{\frac{q \cdot (V + I \times R_S)}{n \times k \times T}} - 1 \right] - \frac{V + I \times R_S}{R_p} \quad (4.5)$$

A (4.5) fornece a curva característica I-V, da célula fotovoltaica para certo nível de irradiância solar e temperatura como se ilustra na Figura 26.

Figura 26 – Curva característica I-V de um módulo



Fonte: (SUNPOWER, 2007)

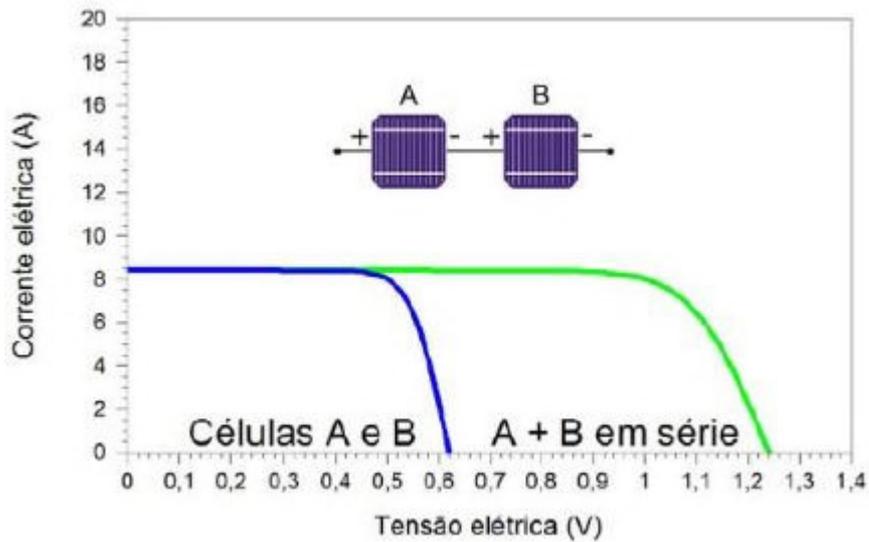
Tabela 9 – Exemplo de características elétricas de painel fotovoltaico marca SunPower modelo 305

Parâmetro	Valor
P _{mp}	305 W
V _{mp}	54,70 V
I _{mp}	5,58 A
V _{oc}	64,2 V
I _{sc}	5,96 A
α	3,5 mA/°C

Fonte: (SUNPOWER, 2007)

A associação série entre módulos é feita conectando o terminal positivo de um módulo ao terminal negativo de outro módulo. Semelhante ao que acontece com células, caso a associação série exista, as tensões de cada módulo são somadas.

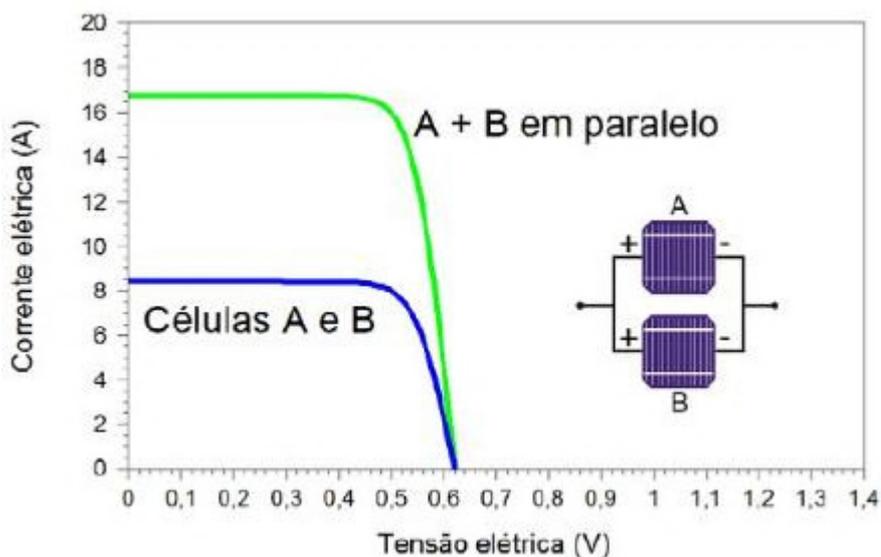
Figura 27 – Associação série de dois módulos



Fonte: CRESESB (2014)

A associação paralela entre módulos é feita conectando o terminal positivo de um módulo ao próprio terminal positivo de outro módulo, seguindo o mesmo raciocínio para os terminais negativos. O efeito da conexão em paralelo é a soma das correntes de cada módulo

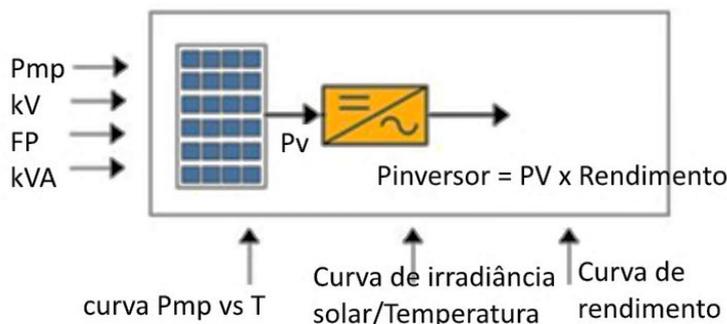
Figura 28 – Associação em paralelo de dois módulos



Fonte: CRESESB (2014)

Para a conexão da USF com o sistema de distribuição empregam-se inversores que convertem a tensão contínua em corrente alternada de forma sincronizada com a tensão da rede. O modelo empregado para a simulação apresenta-se na Figura 29 a partir das características gerais de cada painel.

Figura 29 – Modelo do PVSystem no DiagramaBarra



Fonte: (BASTOS, 2015)

Este modelo foi implementado no DiagramaBarra como elemento *PVSystem* (elemento id 12 da Tabela 7) que possui os painéis fotovoltaicos e o inversor.. O elemento possui comportamento similar ao de uma carga (elemento tipo *Load*) que pode ser controlada, porém com potência negativa, portanto, entregando energia.

O inversor, ou conversor CC-CA, que compõe o *PVSystem* é capaz de atuar em diferentes modos de operação (descrito na Figura 31 e na Tabela 10), sendo possível a sua operação em qualquer região contida nos quatro quadrantes, apresentados na Figura 31. A operação em quatro quadrantes possibilita ao inversor fornecer e absorver potência ativa e reativa do sistema

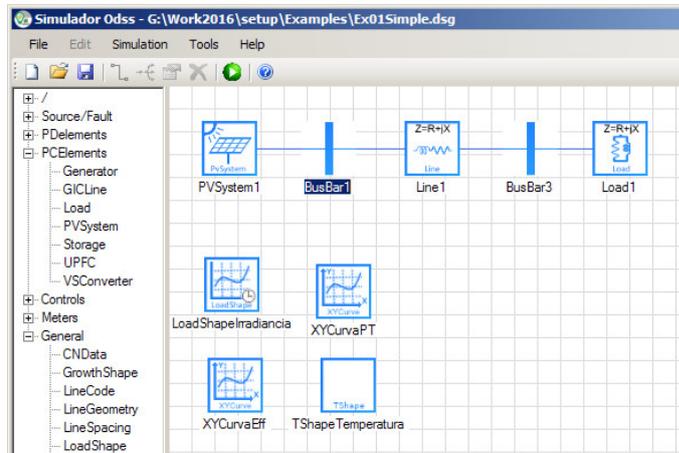
A escolha de um inversor deve ser feita levando em consideração características desejadas de operação como nível de tensão e corrente, rendimento, durabilidade e segurança. Um inversor de qualidade deve atender, dentre outras, as seguintes características (CEPEL/CRESESB, 2014):

- Alta eficiência de conversão;
- Alta confiabilidade e baixa manutenção;
- Operação em uma ampla faixa de tensão de entrada;
- Boa regulação na tensão de saída;
- Forma de onda senoidal com baixo conteúdo harmônico;
- Baixa emissão de ruído audível;

- Segurança tanto para as pessoas quanto para a instalação.
- Modos de operação

As diferentes curvas de comportamento como rendimento, temperatura do painel fotovoltaico e irradiância solar que constam como parâmetros do *PVSystem* correspondem a elementos do tipo *XYCurve*, *TShape* e *LoadShape*, respectivamente. A Figura 30 mostra o diagrama com os elementos descritos.

Figura 30 – Exemplo DiagramaBarra painel fotovoltaico

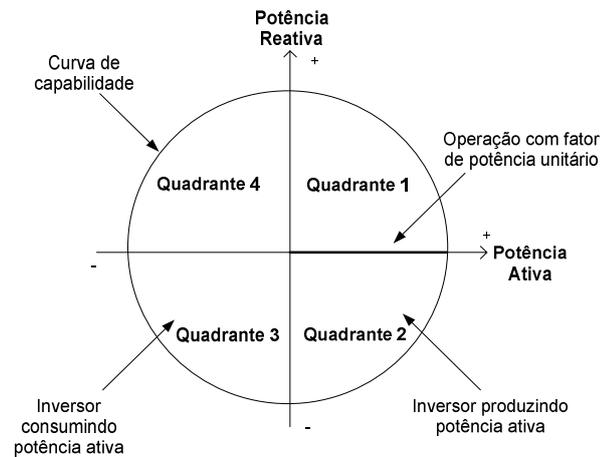


Fonte: Produção do próprio autor

Para realizar a simulação, os parâmetros do painel podem ser obtidos das especificações do fabricante, bastando apenas determinar a potência necessária para ligar os diversos painéis em série e paralelo para atenderem a potência total desejada e determinar um inversor capaz de suportar tensão, corrente e potência dos painéis fotovoltaicos.

Cabe destacar que o inversor possui diferentes modos de operação, sendo possível cobrir todos os quadrantes no espaço de potência como ilustra a Figura 31.

Figura 31 – Quadrantes de operação de um inversor



Fonte: Zilles et al, 2012

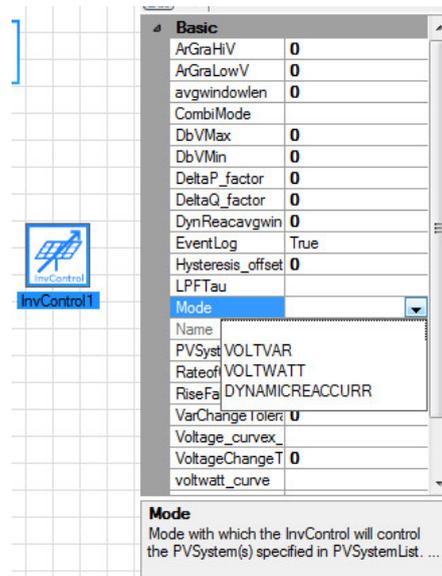
A máxima capacidade de fornecimento ou absorção de potência ativa e/ou reativa pelo inversor é limitada pela sua máxima corrente, que pode ser representada pela sua curva de capacidade da Figura 31. A capacidade de fornecer ou absorver potência reativa em um instante t pode ser quantificado pela Equação (4.6), derivada das relações do triângulo de potência:

$$Q_{(t)}^{max} = \sqrt{(S^{max})^2 - (P_{(t)})^2} \quad (4.6)$$

Assim, para cada instante, a potência reativa depende da máxima potência aparente do inversor e da potência ativa gerada no mesmo instante. Desta forma, os inversores podem injetar potência ativa e reativa simultaneamente em sistemas CA e até mesmo injetar ou absorver apenas potência reativa quando a irradiação solar for nula, ou seja, quando $P(t) = 0$. Portanto, uma usina solar fotovoltaica pode operar como reguladores do fator de potência da rede utilizando diversas estratégias no controle dos inversores.

O programa DiagramaBarra usa o componente *InvController* para o controle dos inversores ligados nos painéis fotovoltaicos. Suas propriedades podem ver-se na Figura 32.

Figura 32 – Propriedades InvController



Fonte: Produção do próprio autor

Os parâmetros podem ser usados em combinação para conseguir diferentes estratégias de controle (Ver Tabela 10 – Diferentes combinações de operação do controle de inversor).

Tabela 10 – Diferentes combinações de operação do controle de inversor

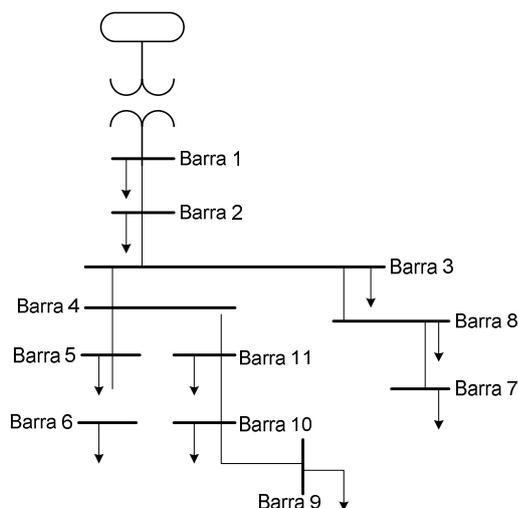
Mode	VV_REFReactivePower	Comportamento inversor
VOLTVAR	VARAVAL_WATTS	Despacha máxima potencia ativa e reativa
VOLTVAR	VARMAX_VARS	Prioriza potência reativa sobre ativa
VOLTVAR	VARMAX_WATTS	Prioriza potência ativa sobre reativa
VOLTWATT	-	Despacho de potencia ativa
DYNAMICREACCURR	-	Modo corrente dinâmico

Fonte: Produção do próprio autor

No estudo publicado analisam-se os efeitos de utilizar o modo de controle VARAVAL_WATTS que despacha máxima potência e reativa, obedecendo os parâmetros impostos pela curva da Figura 36.

As análises de impacto do controle ativo de tensão são realizadas sobre um sistema de distribuição trifásico desbalanceado do CIGRÉ (2009) com 11 barras e tensão fase-fase de 12,47 kV. O Sistema Teste teve como única alteração a exclusão da linha de transmissão que conectava as Barras 8 e 9 para ser transformado em um sistema com topologia radial. Sua carga total é de 4,9+j2,36 MVA. A Figura 33 ilustra o sistema descrito.

Figura 33 – Diagrama do sistema teste

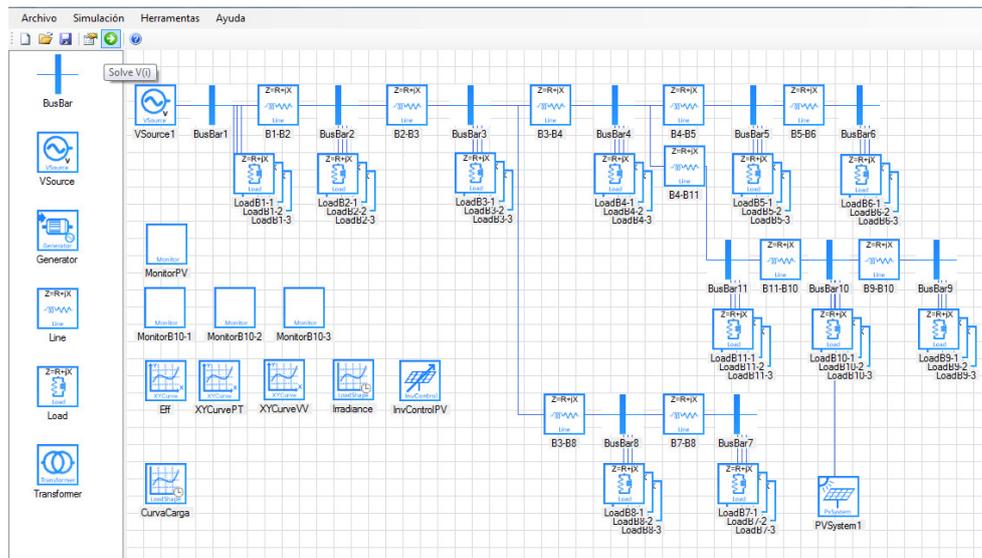


Na publicação, as análises de impacto do controle ativo de tensão são realizadas utilizando análises QSTS, que permitem a resolução sequencial do fluxo de potência de um sistema, ou seja, elas possibilitam não só efetuar o fluxo de potência tradicional em um instante de tempo, mas também soluções consecutivas dele. Para que isto ocorra, como apresentado em (BASTOS, 2015), nas análises QSTS uma solução do fluxo de potência convergido é utilizada como valor inicial para a próxima solução, sendo o intervalo entre cada solução definido de acordo com os dados disponíveis.

A representação do Sistema Teste da Figura 33 na interface gráfica do programa DiagramaBarra é apresentada na Figura 34 sendo possível visualizar também a conexão da USF sob estudo, conectada à Barra 10, pois é nela que há maior redução de perdas elétricas no sistema, obtidas via análises QSTS.

A modo de ilustração, o *script* para OpenDSS que representa este sistema é apresentado no Apêndice A. Fica evidente a diferença entre o *script* com a interface gráfica do DiagramaBarra da Figura 34. Para o aprendiz, a iconografia simbólica tem maior impacto visual, e portanto, um maior valor motivacional.

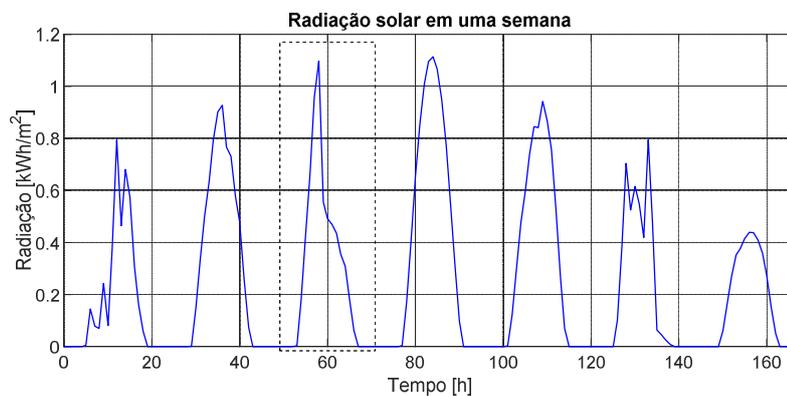
Figura 34 – Sistema teste no DiagramaBarra



Fonte: Produção do próprio autor

Para realizar as análises QSTS no Sistema Teste com uma USF de 1 MWp, foram utilizados dados reais de radiação solar, obtidos para a cidade de Joinville-SC a partir do programa PV*Sol³⁰, com discretizações de tempo de 1 em 1 hora para a primeira semana do ano de 2017. Tais dados são apresentados na Figura 35.

Figura 35 – Curva de radiação solar utilizado



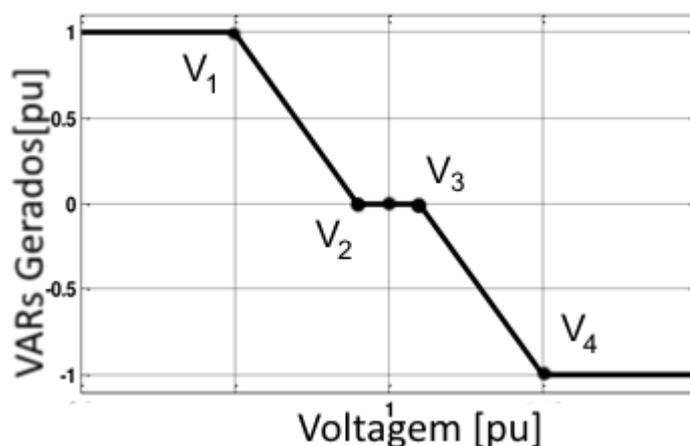
Fonte: Produção do próprio autor

Nas simulações efetuadas, foram consideradas apenas 24 h de simulação, com discretizações de tempo de 1 hora correspondendo ao terceiro dia do ano. Representado pela área em destaque na Figura 35.

³⁰ Disponível em <https://www.valentin-software.com/en/products/photovoltaics>

O controle VOLT-VAR aplicado ao inversor possui lógica de controle similar à dos compensadores estáticos shunt, ou seja, havendo queda de tensão, há injeção de potência reativa e vice-versa. Na Figura 36 é apresentada a curva característica do controle, enfatizando que existe uma zona morta para não ocorrer injeção/absorção de potência reativa se a tensão a ser controlada, geralmente o ponto de conexão, estiver dentro de limites pré-estabelecidos pela norma vigente para o sistema.

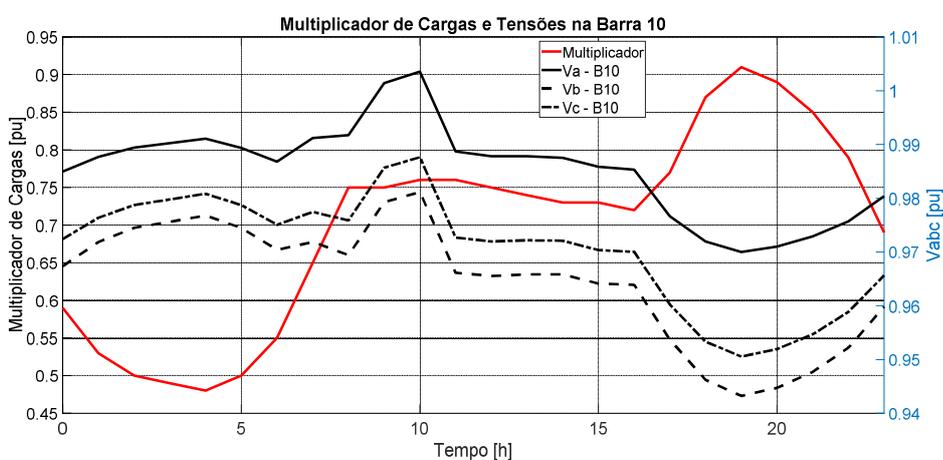
Figura 36 – Curva de ativação do tipo de controle VOLT-VAR



Fonte: RENO, BRODERICK e GRIJALVA (2013)

Os efeitos das variações de carga ao longo de um dia (multiplicador de cargas), da radiação solar e do controle volt-VAr sobre as tensões da Barra 10, onde a USF de 1 MWp foi conectada, são mostrados na Figura 37.

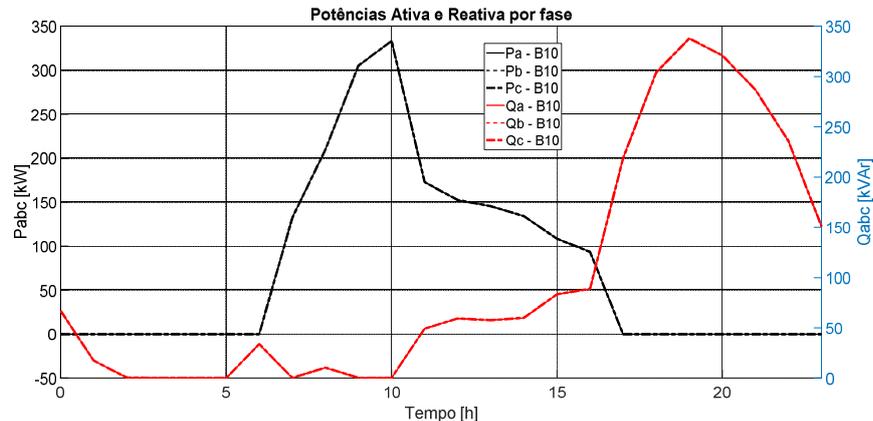
Figura 37 – Curvas de carga e tensão na barra 10



Fonte: Produção do próprio autor

Os perfis de tensão para as fases A, B e C na Barra 10 são diretamente influenciados pelo multiplicador de cargas, escolhido para ser o mesmo para as três fases. Quanto maior o multiplicador de cargas, maior a carga por fase e conseqüentemente maior a queda de tensão.

Figura 38 – Curvas de variação de potência ativa e reativa da USF



Fonte: Produção do próprio autor

Na Figura 38 observa-se que a máxima variações de potência ativa e reativa ao longo do dia, por fase, injetadas pelo inversor de 1 MVA na Barra 10 correspondem a 333 kW na hora 10 e 333 kVAr na hora 19. Nota-se que a potência reativa a cada instante obedece à Equação 4.6 e também à curva característica da Figura 36 – Curva de ativação do tipo de controle VOLT-VAR, onde foram definidos os valores $V_2=0,98$ [pu] e $V_3=1,1$ [pu]. Também se percebe que a potência ativa a cada instante é diretamente proporcional à radiação solar do terceiro dia cujo intervalo de tempo foi destacado na Figura 35 – Curva de radiação solar utilizado. Assim, o controle ativo de tensão pode ser realizado tanto pela injeção de potência ativa como pela injeção de potência reativa via ajuste adequado dos pontos da curva do controle volt-VAr apresentada na Figura 36. É importante notar que embora o sistema seja trifásico e desbalanceado, as injeções de potência ativa e reativa injetadas pelo inversor trifásico são iguais para cada uma das fases.

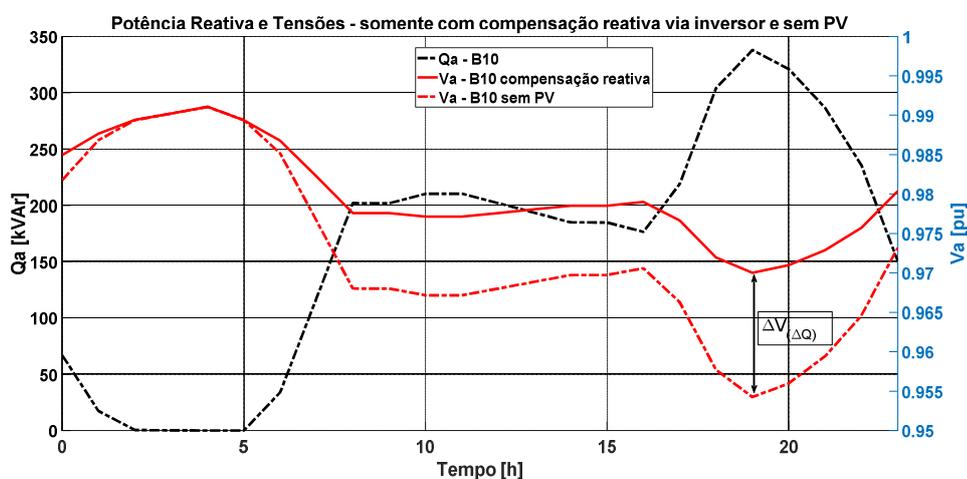
A fim de realizar uma comparação mais justa para inferir se a injeção de potência ativa ou a injeção de potência reativa tem maior influência no controle do perfil de tensão do Sistema Teste, as seguintes configurações foram simuladas:

Caso A: manter o perfil de carga variando de acordo com o multiplicador de cargas, porém considerando um dia com radiação “nula”, por exemplo, um dia

extremamente nublado e chuvoso, onde o inversor da USF atue como um compensador estático de reativos operando no modo de controle VOLT-VAR.

Neste caso as variações de tensão na Barra 10 são apresentadas na Figura 39 em conjunto com o caso sem a conexão da USF. Para a situação de maior carga do sistema, na hora 19, a injeção de 333 kVAr por fase implica em um aumento aproximado de $\Delta V=0,02$ [pu] na tensão da Barra 10, se comparado ao caso sem USF.

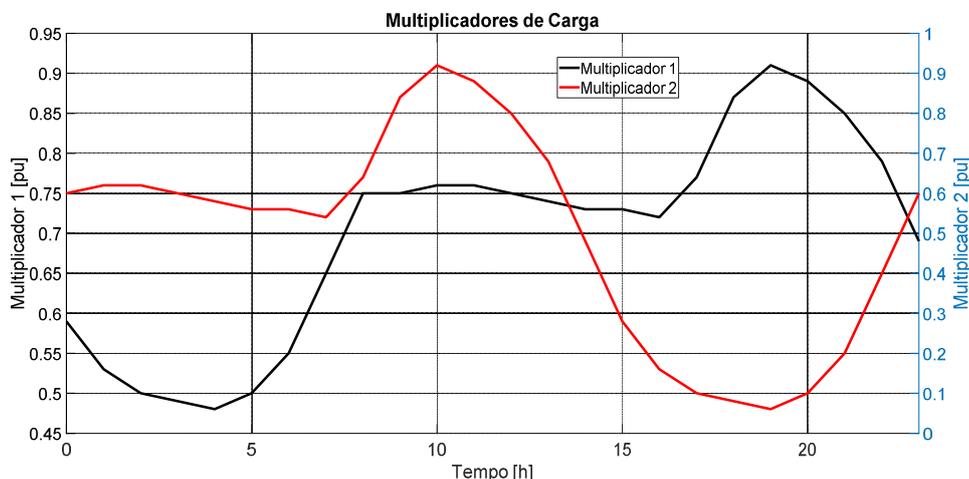
Figura 39 – Variações de tensão na barra 10 com e sem USF para o inversor operando como compensador estático.



Fonte: Produção do próprio autor

Caso B: deslocar o multiplicador de carga para que seu máximo valor ocorra na hora 10, (Multiplicador 2 da Figura 40) onde ocorre o máximo pico de radiação para o dia escolhido, fazendo o inversor operar com fator de potência unitário fixo.

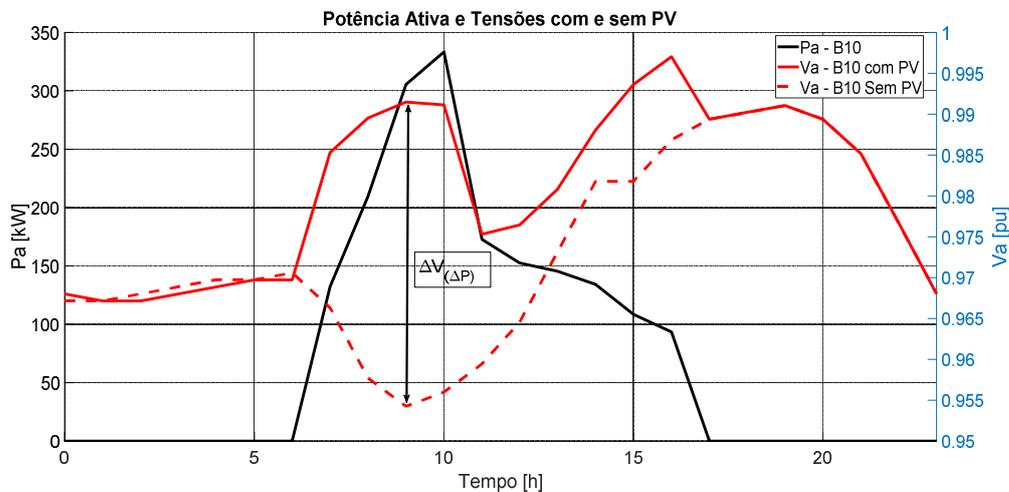
Figura 40 – Multiplicadores de carga utilizados no caso B



Fonte: Produção do próprio autor

As variações de tensão na Barra 10 para o inversor operando com FP=1 fixo e para o caso sem a conexão da USF são apresentadas na Figura 41. Para o maior carregamento do sistema, que ocorre agora na hora 10, a injeção de 333 kW por fase implica em um aumento aproximado de $\Delta V=0,04[\text{pu}]$ na tensão da Barra 10, se comparado ao caso sem USF.

Figura 41 – Variações de tensão com e sem USF para o inversor operando como FP unitário



Fonte: Produção do próprio autor

Ao se comparar os resultados das Figura 40 e Figura 41, para um mesmo valor de pico de carga e mesmas quantidades de injeção de potência reativa e ativa, constatou-se que para este Sistema Teste as variações de tensão são mais sensíveis em relação à injeção de potência ativa do que reativa. Assim, recomenda-se que uma análise de sensibilidade de tensão em uma Barra j , $\partial Q_j / \partial V_j$ e $\partial P_j / \partial V_j$ seja efetuada a priori para verificar a possibilidade de sobre ou sub dimensionamento da potência do inversor a ser utilizado com a finalidade de efetuar o controle ativo de tensão. (WEDDY, ALFA, *et al.*, 2012)

A análise de sensibilidade aplicada a sistemas de transmissão é mais popular se comparada a sua aplicação a sistemas de distribuição. Tal tipo de análise indica que quanto mais a reatância da linha de transmissão diminui e vai se aproximando da resistência, maior deve ser o valor da quantidade de potência reativa injetada para alterar a tensão em um ponto j

$$\frac{\partial P_j}{\partial V_j} < \frac{\partial Q_j}{\partial V_j} \quad (4.7)$$

Analogamente, para redes de distribuição, onde a relação X/R costuma ser baixa, a quantidade de potência reativa injetada para um controle efetivo de tensão deve ser maior. No caso de redes com predominância resistivas, o controle de tensão é mais efetivo via injeção de potência ativa.

$$\frac{\partial P_j}{\partial V_j} > \frac{\partial Q_j}{\partial V_j} \quad (4.8)$$

Uma alternativa mais efetiva para o controle ativo de tensão para este Sistema Teste seria o controle VOLT-WATT. Porém este tipo de controle é mais útil para evitar sobretensões, pois quando esta situação acontece, a potência ativa injetada na rede é reduzida ou até mesmo pode ocorrer o desligamento da USF se a mesma não possuir um sistema de armazenamento de energia. Caso possua um sistema de armazenamento, este “excesso” de potência ativa que causaria sobretensão é usado para carregar o sistema de armazenamento de energia. Logo, para minimizar subtensões, em períodos de radiação baixa ou nula, o controle VOLT-WATT só seria mais efetivo que o controle VOLT-VAR se a USF estivesse acoplada a um sistema de armazenamento de energia, que possibilitaria a injeção de potência ativa à rede pelo inversor via armazenador de energia.

Conclui-se que efetuar análises de sensibilidade para cada tipo de sistema (transmissão ou distribuição) é um fator decisivo para a escolha do modo de controle do inversor no uso de USF.

4.4 APLICAÇÃO DO DIAGRAMA BARRA NO ANÁLISE DE PERTURBAÇÕES /SENSIBILIDADE

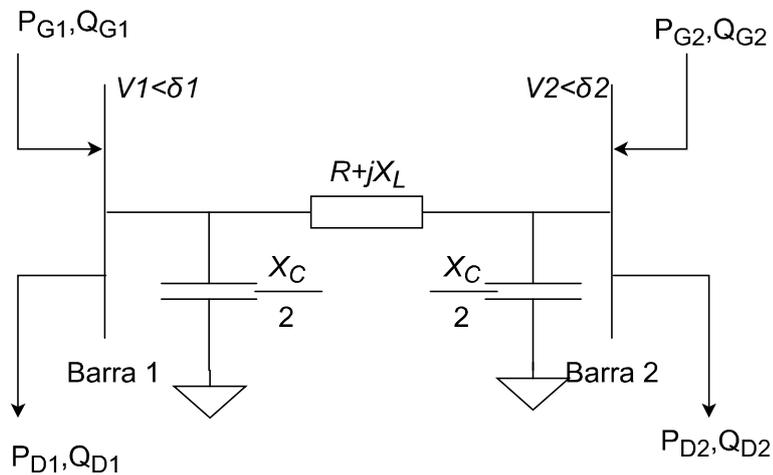
No trabalho de análise de perturbações ou da sensibilidade de sistemas de energia estuda-se como influi na rede a presença de pequenas variações na geração e na demanda. A importância deste tipo de estudo reside na determinação de como afeta uma variável no equilíbrio do fluxo de potência. Isso ajuda a

estabelecer critérios para projetos de sistemas de controle e avaliar a estabilidade de sistemas diante de variações na demanda e geração.

A análise parte do modelagem de um esquema de duas barras e uma linha de transmissão. Com base em (ELGERD, 1976), a seguir apresenta-se os equacionamentos que determinam as relações entre as variáveis de um sistema de energia quando frente a perturbações. Posteriormente se compara com o resultado no software DiagramaBarra utilizando a ferramenta para gerar curvas e gráficos

O sistema de equações que descreve o sistema da Figura 42 está definido pelas equações de potência (4.9) e equação de equilíbrio(4.10).

Figura 42 – Sistema de duas barras e uma linha



Fonte: Adaptado (ELGERD, 1976)

$$\begin{cases} S_1 = P_1 + jQ_1 \\ S_2 = P_2 + jQ_2 \end{cases} \quad (4.9)$$

$$\begin{cases} \frac{S_1^*}{V_1} = Y_{sh} + \frac{V_1 - V_2}{Z_{ser}} \\ \frac{S_2^*}{V_2} = Y_{sh} + \frac{V_2 - V_1}{Z_{ser}} \end{cases} \quad (4.10)$$

Sendo

$$Y_{sh} = \frac{j}{X_C} \quad (4.11)$$

$$Z_{ser} = R + jX_L \quad (4.12)$$

Define-se o fator de perdas pela equação (4.13)

$$\alpha = \frac{R}{X_L} \ll 1 \quad (4.13)$$

Nesta condição pode-se escrever (4.12) na forma complexa:

$$Z_{ser} \approx X_L e^{j(\pi/2 - \alpha)} \quad (4.14)$$

Usando a notação da tensão na forma complexa tem-se:

$$\begin{cases} V_1 = |V_1| e^{j\delta_1} \\ V_2 = |V_2| e^{j\delta_2} \end{cases} \quad (4.15)$$

Substituindo (4.11) (4.12) (4.13) e (4.14) em (4.9) e (4.10), agrupando parte real e imaginária, e separando a geração P_G do consumo P_D se obtém o sistema da equação (4.16).

$$\begin{aligned} P_{G1} - P_{D1} - \frac{|V_1|^2}{X_L} \sin \alpha + \frac{|V_1||V_2|}{X_L} \sin[\alpha - (\delta_1 - \delta_2)] &= 0 \\ P_{G2} - P_{D2} - \frac{|V_2|^2}{X_L} \sin \alpha + \frac{|V_1||V_2|}{X_L} \sin[\alpha + (\delta_1 - \delta_2)] &= 0 \\ Q_{G1} - Q_{D1} + \frac{|V_1|^2}{X_C} - \frac{|V_1|^2}{X_L} \cos \alpha + \frac{|V_1||V_2|}{X_L} \cos[\alpha - (\delta_1 - \delta_2)] &= 0 \\ Q_{G2} - Q_{D2} + \frac{|V_2|^2}{X_C} - \frac{|V_2|^2}{X_L} \cos \alpha + \frac{|V_1||V_2|}{X_L} \cos[\alpha + (\delta_1 - \delta_2)] &= 0 \end{aligned} \quad (4.16)$$

Usando a notação vetorial, se tem os vetores de dimensão $[2n]$, sendo n o número de barras.

$$\mathbf{p} = \begin{bmatrix} P_{Di} \\ Q_{Di} \\ \vdots \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \delta_i \\ |V|_i \\ \vdots \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} P_{Gi} \\ Q_{Gi} \\ \vdots \end{bmatrix} \quad (4.17)$$

O sistema de equações (4.16) pode ser descrito na forma de funções

$$f_k(\mathbf{p}, \mathbf{x}, \mathbf{u}) = 0, \quad \text{com } k = 1, \dots, 2n \quad (4.18)$$

Usando os valores iniciais para caso de regime permanente

$$f_i^0 = f_i(x^0, u^0, p^0) = 0 \quad (4.19)$$

No caso da presença de perturbações o sistema vetorial (4.19) se transforma em (4.20)

$$f_i^0 = f_i(x^0 + \Delta x, u^0 + \Delta u, p^0 + \Delta p) = 0 \quad (4.20)$$

Utilizando a expansão numa série de Taylor de f_i em torno do valor de equilíbrio (4.19) e se estima que as perturbações sejam pequenas tal que possa desprezar os infinitésimos de ordem superior.

$$f_i(x^0 + \Delta x, u^0 + \Delta u, p^0 + \Delta p) = f_i(x^0, u^0, p^0) + \sum_{j=1}^{2n} \frac{\partial f_i}{\partial x_j} \Delta x_j + \sum_{j=1}^{2n} \frac{\partial f_i}{\partial u_j} \Delta u_j + \sum_{j=1}^{2n} \frac{\partial f_i}{\partial p_j} \Delta p_j \approx 0 \quad (4.21)$$

Substituindo a condição de regime permanente (4.20) em (4.21) obtém-se a expressão para cada componente.

$$\sum_{j=1}^{2n} \frac{\partial f_i}{\partial x_j} \Delta x_j + \sum_{j=1}^{2n} \frac{\partial f_i}{\partial u_j} \Delta u_j + \sum_{j=1}^{2n} \frac{\partial f_i}{\partial p_j} \Delta p_j = 0, \quad \text{sendo } i = 1, \dots, 2n \quad (4.22)$$

Usando os vetores de perturbação (4.23)

$$\Delta x = \begin{bmatrix} \Delta \delta_i \\ |\Delta V|_i \\ \vdots \end{bmatrix}, \quad \Delta p = \begin{bmatrix} \Delta P_{Di} \\ \Delta Q_{Di} \\ \vdots \end{bmatrix}, \quad \Delta u = \begin{bmatrix} \Delta P_{Gi} \\ \Delta Q_{Gi} \\ \vdots \end{bmatrix} \quad (4.23)$$

Podemos expressar (4.22) na forma matricial $2n \times 2n$ do sistema formando (4.24).

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_{2n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{2n}}{\partial x_1} & \dots & \frac{\partial f_{2n}}{\partial x_{2n}} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_{2n} \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \dots & \frac{\partial f_1}{\partial u_{2n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{2n}}{\partial u_1} & \dots & \frac{\partial f_{2n}}{\partial u_{2n}} \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \vdots \\ \Delta u_{2n} \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial p_1} & \dots & \frac{\partial f_1}{\partial p_{2n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{2n}}{\partial p_1} & \dots & \frac{\partial f_{2n}}{\partial p_{2n}} \end{bmatrix} \begin{bmatrix} \Delta p_1 \\ \vdots \\ \Delta p_{2n} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.24)$$

Expressado de outra forma

$$J_x \Delta x + J_u \Delta u + J_p \Delta p = 0 \quad (4.25)$$

Tem-se que as matrizes de derivadas parciais são chamadas de matrizes jacobianas J_i $i=x,u,p$.

Solucionando (4.25) para Δx

$$\Delta x = -J_x^{-1} J_u \Delta u - J_x^{-1} J_p \Delta p = S_u \Delta u + S_p \Delta p \quad (4.26)$$

Se definem as chamadas matrizes de sensibilidade

$$\begin{aligned} S_u &= -J_x^{-1} J_u \\ S_p &= -J_x^{-1} J_p \end{aligned} \quad (4.27)$$

Desta forma, o valor do vetor de estado Δx fica definido em função do vetor de injeções de potência ativa e reativa Δu e do vetor de demanda Δp .

No caso do exemplo 7.2 de (ELGERD, 1976), apresentam-se as seguintes condições:

$$\begin{aligned} |V_1| &= 1[pu], & \delta_1 &= 0, \\ P_{D1} &= P_{D2} = 20[pu], & Q_{D1} &= Q_{D2} = 10[pu], & P_{G2} &= 20[pu] \end{aligned} \quad (4.28)$$

Com $V_{\text{base}} = 120 \text{ KV}$, $P_{\text{base}} = 50 \text{ MW}$, $X_L = 0,1$

Assumindo que Barra1 como barra de referência no sistema da Figura 42, as incognitas são $|V_2|$, δ_2 e Q_{G1} . Como a condição de $P_{D2} = P_{G2} = 20 [\text{pu}]$, a tensão e fase é igual nas duas barras, desprezando $X_c = \infty$ e $\alpha = 0$.

Nestas condições pode expressar-se os valores de p_i em função de x_i , ficando $p_1 = 20x_2^2$, $p_2 = 10x_2^2$, $p_3 = 20x_4^2$, $p_4 = 10x_4^2$. Desta forma as funções f_i definidas pelo sistema de equações (4.16), com $\alpha = 0$, se transformam em (4.29).

$$\begin{aligned}
 f_1(\mathbf{x}, \mathbf{u}, \mathbf{p}) &= u_1 - 20x_2^2 + \frac{x_2x_4}{X_L} \sin(-x_1 + x_3) = 0 \\
 f_2(\mathbf{x}, \mathbf{u}, \mathbf{p}) &= u_3 - 20x_4^2 + \frac{x_2x_4}{X_L} \sin(x_1 - x_3) = 0 \\
 f_3(\mathbf{x}, \mathbf{u}, \mathbf{p}) &= u_2 - 10x_2^2 - \frac{x_2^2}{X_L} + \frac{x_2x_4}{X_L} \cos(-x_1 + x_3) = 0 \\
 f_4(\mathbf{x}, \mathbf{u}, \mathbf{p}) &= u_4 - 10x_4^2 - \frac{x_4^2}{X_L} + \frac{x_2x_4}{X_L} \cos(x_1 - x_3) = 0
 \end{aligned} \tag{4.30}$$

Com $P_{Gi} = P_{Di}$, então $x_1 = x_3$. Portanto os Jacobianos ficam definidos pela expressão (4.31).

$$\mathbf{J}_x = \begin{bmatrix} \frac{x_2x_4}{X_L} & -40x_2 & -\frac{x_2x_4}{X_L} & 0 \\ -\frac{x_2x_4}{X_L} & 0 & -\frac{x_2x_4}{X_L} & -40x_4 \\ 0 & -20x_2 - \frac{x_2}{X_L} & 0 & \frac{x_2}{X_L} \\ 0 & \frac{x_4}{X_L} & 0 & -20x_4 - \frac{x_2}{X_L} \end{bmatrix} = \begin{bmatrix} 10 & -40 & -10 & 0 \\ -10 & 0 & -10 & -40 \\ 0 & -30 & 0 & 10 \\ 0 & 10 & 0 & -30 \end{bmatrix} \tag{4.31}$$

$$\mathbf{J}_u = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{J}_p = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{4.32}$$

E a matriz de sensibilidade \mathbf{S}_u

$$\mathbf{S}_u = -\mathbf{J}_x^{-1} \mathbf{J}_u = \begin{bmatrix} -50 & 50 & 50 & -50 \\ 0 & 37,5 & 0 & 12,5 \\ 50 & -100 & 50 & -100 \\ 0 & 12,5 & 0 & 37,5 \end{bmatrix} \times 10^{-3} \tag{4.33}$$

Define-se a Barra 1 como uma barra de referencia, ou seja. $\Delta\delta_1 = \Delta x_1 = 0$. Realizando operações fila-coluna na matriz (4.33) se pode eliminar a primeira fila da matriz.

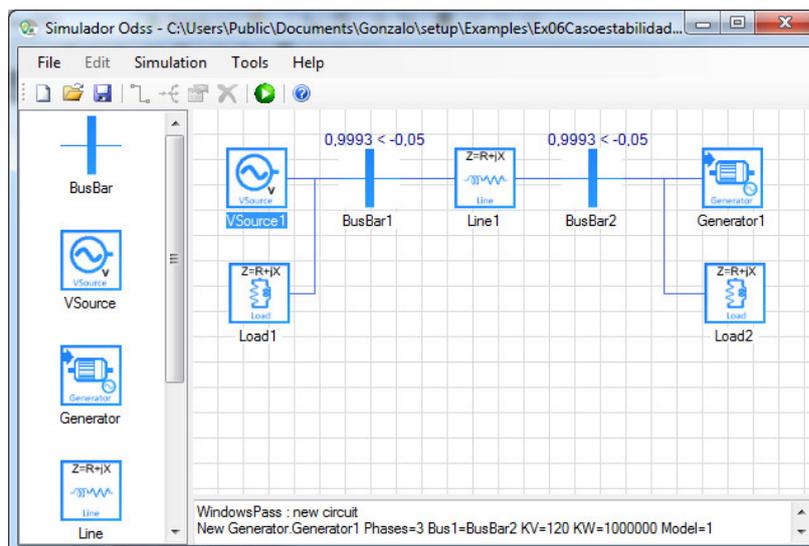
$$\mathbf{S}_u = \begin{bmatrix} 0 & 37,5 & 0 & 12,5 \\ 0 & -50 & 100 & -150 \\ 0 & 12,5 & 0 & 37,5 \end{bmatrix} \times 10^{-3} \quad (4.34)$$

$$\begin{aligned} \Delta V_1 &= 0,0375 \cdot \Delta Q_{G1} + 0,0125 \cdot \Delta Q_{G2} \\ \Delta\delta_2 &= -0,05 \cdot \Delta Q_{G1} + 0,1 \cdot \Delta P_{G2} - 0,15 \cdot \Delta Q_{G2} \\ \Delta V_2 &= 0,0125 \cdot \Delta Q_{G1} + 0,0375 \cdot \Delta Q_{G2} \end{aligned} \quad (4.35)$$

Sendo assim, de acordo com (4.35) o ângulo δ_2 diminui na medida que adicionamos potência reativa, e aumenta quando adicionamos potência ativa.

Montando o sistema da Figura 42 no programa DiagramaBarra como apresentado na.Figura 43. Foram configurados os parâmetros como mencionado anteriormente e seguindo o exemplo 7.2 de (ELGERD, 1976), o resultado da simulação confere com a teoria.

Figura 43 – Diagrama do sistema na qual se realizou análise de sensibilidade



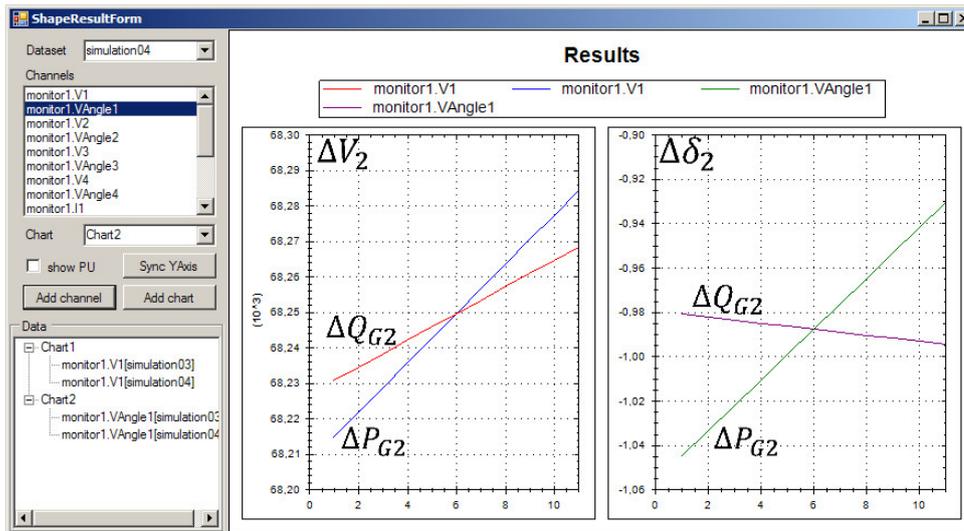
Fonte: Produção do próprio autor.

Adicionando as seguintes curvas paramétricas :

- Caso a) variação de 95% a 105% no valor nominal de P_{G2} ;
- Caso b) variação de 95% a 105% no valor nominal de Q_{G2} .

Simularam-se as curvas comprovando-se a equação de sensibilidade (4.35).

Figura 44 – Interface gráfica desenvolvida com os resultados da análise de sensibilidade



Fonte : Produção do próprio autor

A solução do problema usando este método pode ser aplicado na solução de um problema de n barras para analisar a influência da variação de cargas nas variáveis de estado.

No Apêndice B, se incluem exemplos de atividades pedagógicas baseadas no programa DiagramaBarra para o tutor usar em aula ou como tarefas para os aprendizes. Os mesmos exemplos encontram-se disponíveis para descarga no site *SourceForge*.

5 CONCLUSÕES

5.1 CONCLUSÕES E CONSIDERAÇÕES FINAIS

Este trabalho teve como principal objetivo o desenvolvimento de uma interface gráfica de simulação de redes de energia para aplicação no ensino, possível de ser usado no ambiente profissional, com fundamentos na filosofia de *software* livre.

O projeto realizado cumpriu o objetivo incluindo-se várias ferramentas de apoio para o aprendizado: capacidade de internacionalização para uso de múltiplos idiomas, uma interface gráfica com iconografia simbólica e ajuda contextual. Apresentaram-se aplicações, entre elas uma análise da influência de usina fotovoltaica em sistemas de energia com distintos modos de funcionamento.

No aspecto teórico e de fundamentação que existem varias oportunidades na área de simulação de redes para o desenvolvimento de produtos para aplicações de ensino, na área de otimização de algoritmos, na aplicação de novas estratégias de controle e de novas tecnologias, como a simulação híbrida e a co-simulação.

Estudou-se a atualidade dos paradigmas para descobrir novas tendências no ensino aplicado. Descobriu-se que na área da engenharia, existem vários estudos que evidenciam a carência de estratégias pedagógicas atualizadas, e que estão aplicando-se planos de melhoramento do corpo docente com avaliações e cursos de aperfeiçoamento.

No desenvolvimento, comprovou-se a possibilidade de programar *software* de diagramas usando bibliotecas gratuitas voltadas para criação de interfaces. Ainda que tais bibliotecas não fossem utilizadas diretamente na programação de DiagramaBarra, serviram de referência para gerar-se uma nova biblioteca. A facilidade de uso da plataforma *.NET Framework* demonstrou a possibilidade de implementar um programa que pode executar em qualquer sistema operacional.

A incorporação de ferramentas adicionais no software DiagrabaBarra, dá um valor agregado ao produto que assim apresenta vantagens para o uso no ensino.

5.2 PROPOSTAS PARA TRABALHOS FUTUROS

O trabalho de criar um novo programa de código aberto para a simulação de sistemas de energia abre a possibilidade de novos trabalhos de melhoria da interface. Alguns exemplos de melhoria da interface são:

- Expansão da plataforma para incluir sistema operacional Android.
- Expansão para gerar equações do circuito a partir dos diagramas.
- Adicionar ferramentas de análise de sensibilidade.

Tecnicamente, seria possível desenvolver ferramentas similares com conteúdos armazenados na nuvem. Mediante a plataforma *.NET Framework* se possibilita o uso do programa DiagramaBarra como uma biblioteca para novas aplicações.

No âmbito educativo, uma proposta de trabalho futuro seria gerar um *framework* aberto que cumprisse as diretrizes de objeto de aprendizado e fosse compatível com os diferentes programas gerados nas universidades. Tal plataforma permitiria definir normas de comunicação e bibliotecas para ser incorporadas em todo tipo de software com finalidades pedagógicas.

Nas técnicas de aprendizado, podem usar-se os exemplos aplicados como base para outro software de uso acadêmico. Sugere-se realizar o acompanhamento e avaliação do DiagramaBarra através de métricas de desempenho na aprendizagem. Também é possível realizar uma pesquisa de necessidades que não estão cobertas pelos *softwares* usados na tarefa educativa.

REFERENCIAS

AKOUE, B. et al. ***Starter Teachers a Methodology Course for the Classroom***. Londres: British Council, 2014.

ANDERSON, P. H.; LAWTON, L. *Business simulations and cognitive learning*. ***Simulation & Gaming***, v. 40, n. 2, 2009.

ANEEL. **Nota Técnica nº 0057/2014-SRD/ANEEL**. ANEEL. [S.I.]. 2014.

ARANHA, M. L. **História da Educação**. 2. ed. São Paulo: Moderna, 1996.

BASTOS, C. **Estudo dos impactos de um sistema fotovoltaico conectado à rede elétrica usando análise QSTS**. 2015, Dissertação (Mestrado em engenharia elétrica) - UDESC, Joinville. 2015.

BAZZO, W. A. **Desafios da educação em engenharia: vocação, formação, exercício profissional, experiências metodológicas e proposições**. Brasília, DF: Abenge, 2012.

BAZZO, W. A. **Ciência, tecnologia e sociedade: e o contexto da educação tecnológica**. Florianópolis: Ed. da UFSC, 2014.

BROADWATER, R. et al. *Application programmer interface for the EPRI*. ***IEEE Transactionson Power Systems***, 10, n. 1, 1995.

BUYYA, R. ***Cloud computing: The next revolution in information technology***. 2010 *First International Conference On Parallel, Distributed and Grid Computing (PDGC 2010)*. Solan, India: IEEE. 2010.

CANTO, A. B.; MÜLLER, T. J.; LIMA, J. V. Estilos de Aprendizagem: vale a pena investir? **RENTE - Revista Novas Tecnologias na Educação, CINTED-UFRGS**, Porto Alegre, v. 13, p. 1-10, 2015.

CCT UDESC, 2018. Disponível em: <<https://www.udesc.br/cct/home>>. Acesso em: 10 fevereiro 2019.

CHOUHAN, S. ***Agent based modeling of power distribution systems***. West Virginia University. Morgantow, WV. 2009.

CIGRE. ***Benchmark Systems for Network Integration of Renewable and Distributed Energy Resources***. CIGRE Task Force. Final Report. 2009.

CRESESB. **Manual de engenharia para sistemas fotovoltaicos**. Rio de Janeiro-RJ: CRESESB, 2014.

DAVOK, D. F.; BERNARD, R. P. Avaliação dos índices de evasão nos cursos de graduação da Universidade do Estado de Santa Catarina - UDESC. **Avaliação**, Campinas, v. 21, nº2, Julho 2016.

DIAZ BORDENAVE, J. E.; PEREIRA, A. M. **Estratégias de ensino-aprendizagem**. 33. ed. Petrópolis: Vozes, 2015.

DUSSEL, E. **La pedagógica latinoamericana**. Segunda. ed. Bogotá: Editorial Nueva América, 1980.

DZAFIC, I.; GLAVIC, M.; TESNJAK, S. **An object-oriented graphical package for power system simulation and analysis**. *12th IEEE Mediterranean Electrotechnical Conference*. Dubrovnik, Croatia: IEEE. 2004.

ELENKOVA, M. Z. et al. **A Simulation Platform for Smart Microgrids in University Campuses**. *52nd International Universities' Power Engineering Conference (UPEC 2017)*. Heraklion: IEEE. 2017.

ELGERD, O. I. **Introdução à teoria de sistemas de energia elétrica**. São Paulo: McGraw-Hill, 1976.

EPRI. Disponível em: <<http://smartgrid.epri.com>>. Acesso em: 10 fevereiro 2018.

EPRI. **New User Primer - The Open Distribution System Simulator**. EPRI. [S.I.]. 2012.

GIL, A. C. **Metodologia do ensino superior**. 4. ed. São Paulo: Atlas, 2005.

GRAINGER, J.; STEVENSON, W. D. **Power system analysis**. New York-NY: McGraw-Hill, 1994.

HARTMANN, B. et al. **Enhancing model interchangeability for powerflow studies: An example of a new Hungarian network model in powerfactory and eAsimov**. *Proceedings 31st European Conference on Modelling and Simulation - ECMS*. Budapest: [s.n.]. 2017.

HICKSON, R. **Aprenda a programar em C, C++ e C#**. 2. ed. rev. e atual. ed. Rio de Janeiro: Campus, 2005.

JDEED, M.; SHARMA, E.; ELMENREICH, W. **Smart grid modeling and simulation - Comparing GridLAB-D and RAPSIm via two Case studies**. *IEEE International Energy Conference (ENERGYCON)*. Limassol (Chipre): IEEE. 2018.

KONDORO, A. et al. **Simulation tools for a smart micro-grid: comparison and outlook**. 21st FRUCT. Helsinki: *Finnish-Russian University Cooperation in Telecommunications*. 2017.

KRISHNAMURTHY, D. *psst: An open-source power system simulation toolbox in Python*. North American Power Symposium (NAPS). Denver, CO: IEEE. 2016.

KYESSWA, M. et al. **Generator Model Extension for Higher Accuracy Simulation of Power System Transients in OpenModelica**. Fourth International Conference on Mathematics and Computers in Sciences and in Industry (MCSI). Corfu: IEEE. 2017.

LENZA, C. **Descolonización educativa en el Estado Plurinacional de Bolivia: Desafíos y dilemas**. 2015, Tese (Doutorado) - Universidade de Bérghamo, Bérghamo. 2015.

LODER, L. L. **Epistemologia versus Pedagogia: o locus do professor de engenharia**. 2002, Dissertação (Mestrado) UFRGS, Porto Alegre-RS. 2002.

LODER, L. L. **Engenheiro em formação: o sujeito da aprendizagem e a construção do conhecimento em engenharia elétrica**. 2009, Tese (Doutorado) - UFRGS, Porto Alegre-RS. 2009.

MATURANA, H. **Cognição, ciência e vida cotidiana**. Belo Horizonte-MG: Ed. UFMG, 2001. 203 p.

MILANO, F.; VANFRETTI, L. **State of the art and future of OSS for power systems**. IEEE Power & Energy Society General Meeting. Calgary, AB: IEEE. 2009.

MILANO, F.; ZHOU, M.; HOU, G. **Open model for exchanging power system data**. Power & Energy Society General Meeting (PES). [S.I.]: IEEE. 2009.

MOFFET, M.-A.; SIROIS, F.; BEAUVAIS, D. **Review of Open Source Code Power Grid Simulation Tools for Long Term Parametric Simulation**. CanmetENERGY technical report. Quebec, p. 137. Julho, 2011.

MONO. Disponível em: <<https://www.mono-project.com/>>. Acesso em: 10 fevereiro 2019.

MONTENEGRO, D.; DUGAN, R. C. **OpenDSS and OpenDSS-PM Open Source Libraries**. IEEE Workshop on Power Electronics and Power Quality Applications. Bogotá: IEEE. 2017.

MOREIRA, M. A.; MASSONI, N. T. **Epistemologias do século XX**: Popper, Kuhn, Lakatos, Laudan, Bachelard, Toulmin, Feyerabend, Maturana, Bohm, Bunge, Prigogine, Mayr. São Paulo: EPU, 2011.

MURAD, M. A. A. et al. **Enhancing engineering studies in developing countries using OpenModelica**. 4th International Conference on Advances in Electrical Engineering (ICAEE). Dhaka: IEEE. 2017.

OLIVA, A.; MELO, T. E. D. **A revolução do software livre**. Manaus: Comunidade Sol Software Livre, 2009. 366 p.

ORACLE , 2018. Disponível em: <<https://www.oracle.com>>. Acesso em: 10 fevereiro 2019.

PIAGET, J. **A epistemología genética**. 2. ed. São Paulo: Abril Cultural, 1983.

PORCHACKER, M. . S. A. . E. W. **Simulating the Smart Grid**. PowerTech. Grenoble: IEEE. 2013.

POYRAZOGLU, G.; OH, H. **The Impacts of Electric Power Market Simulation on Engineering Education**. IEEE/PES Transmission and Distribution Conference and Exposition (T&D). Dallas, TX: IEEE. 2016.

REITORIA, E. A. **Relatório de gestão UDESC**. UDESC. Florianópolis-SC. 2018;2017;2016;2015.

RENO, M.; BRODERICK, R.; GRIJALVA, S. **Smart Inverter Capabilities for Mitigating Over-Voltage on Distribution Systems with High Penetrations of PV**. IEEE Photovoltaic Specialists Conference. Tampa-FL: IEEE. 2013.

ROSSETTO, E. A educação a luz do pensamento de Maturana. **Revista Educação Especial**, Santa Maria-RS, v. 21, n. 32, 2008.

SCHÖN, D. A. **Educando o profissional reflexivo: um novo design para o ensino e a aprendizagem**. Porto Alegre: Artmed, 2000.

SOURCEFORGE. Disponível em: <<http://sourceforge.net>>. Acesso em: 10 fevereiro 2019.

SPÍNDOLA, M. M. Transversalidade dentro da sala de aula do Ensino Superior. **Especialize**, Instituto de Pós-Graduação - IPOG, Goiânia-GO, Julho 2017.

STIFTER, M. et al. **Co-Simulation of Components, Controls and Power Systems based on Open Source Software**. IEEE PES General Meeting. Vancouver-BC: IEEE. 2013.

SUNPOWER. *SunPower 305 datasheet*, 2007. Disponível em: <<http://www.sunpower.com>>. Acesso em: 12 nov 2018.

THURNER, L. et al. ***Pandapower - An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems.*** *IEEE Transactions on Power Systems.* [S.l.]: IEEE. 2018.

TURTIAINEN, A. et al. ***Simulation Game in Teaching Electric Economics.*** *IEEE/PES Transmission and Distribution Conference and Exhibition.* Yokohama: IEEE. 2002.

VISUALSTUDIO. Disponível em: <<https://visualstudio.microsoft.com>>. Acesso em: 10 fevereiro 2019.

WEDDY, B. M. et al. ***Electric Power Systems.*** 5ª Edição. ed. West Sussex-UK: John Wiley & Sons, 2012.

ZHANGBIAO, S.; SHANAN, Z. ***Online virtual experiment of power electrical devices based on XML and modelica.*** *10th IEEE International Conference on Control and Automation (ICCA).* Hangzhou: IEEE. 2013.

ZILLES, R. et al. ***Sistemas fotovoltaicos conectados à rede elétrica.*** [S.l.]. 2012.

APÊNDICE A

Quadro 1 – Representação do Sistema Teste da Figura 31 via scripts no OpenDSS.

```

Clear
New object=circuit.PV
~ bus1=B1 Basekv=12.47 pu=1.03 x1r1=0.4
//curva Potencia versus Temperatura, P-T, para uma irradiancia igual a 1000W/m²
New XYCurve.PvsT npts=4 xarray=[0 25 50 75] yarray=[1.1 1.0 0.9 0.8]
//curva Eficiencia do Inversor versus Potência, EffCurve
New XYCurve.Eff npts=4 xarray=[.1 .2 .4 1.0] yarray=[.86 .9 .93 .97]
// niveis de irradiancia ao longo de um ddia
New Loadshape.Irradiancia 24 1.0
~ mult=[file='Joinville.csv']
plot loadshape object=Irradiancia
//*****
New Tshape.MyTemp npts=24 interval=1
~temp=[25, 25, 25, 25, 25, 25, 25, 25, 35, 40, 45, 50 60 60 55 40 35 30 25
25 25 25 25 25]
//*****
New Loadshape.carga 24 1.0
~ mult=[file='carga1hDesloc.csv']
plot loadshape object=carga
//parametros de linha em ohm @ 12.47 kV
new line.linha1 phases=3 bus1=B1 bus2=B2 r1=0.3460 x1=0.7776 r1=0.3460 x1=0.7776
c1=0 c0=0 Length=1
new line.linha2 phases=3 bus1=B2 bus2=B3 r1=0.5190 x1=0.6480 r1=0.5190 x1=0.6480
c1=0 c0=0 Length=1
new line.linha3 phases=3 bus1=B3 bus2=B4 r1=0.6920 x1=0.3974 r1=0.6920 x1=0.3974
c1=0 c0=0 Length=1
new line.linha4 phases=3 bus1=B3 bus2=B8 r1=1.3840 x1=0.8424 r1=1.3840 x1=0.8424
c1=0 c0=0 Length=1
new line.linha5 phases=3 bus1=B4 bus2=B5 r1=0.8650 x1=0.3628 r1=0.8650 x1=0.3628
c1=0 c0=0 Length=1
new line.linha6 phases=3 bus1=B4 bus2=B11 r1=1.9030 x1=0.3240 r1=1.9030 x1=0.3240
c1=0 c0=0 Length=1
new line.linha7 phases=3 bus1=B5 bus2=B6 r1=1.0380 x1=0.9979 r1=1.0380 x1=0.9979
c1=0 c0=0 Length=1
new line.linha8 phases=3 bus1=B7 bus2=B8 r1=1.3840 x1=1.0843 r1=1.3840 x1=1.0843
c1=0 c0=0 Length=1
////////////////////////////////////new line.linha9 phases=3 bus1=B8 bus2=B9 r1=1.5570 x1=0.2074

```

Quadro 1 – Representação do Sistema Teste da Figura 31 via scripts no OpenDSS.

```

r1=1.5570 x1=0.2074 c1=0 c0=0 Length=1
new line.linha10 phases=3 bus1=B9 bus2=B10 r1=1.7300 x1=0.5184 r1=1.7300 x1=0.5184
c1=0 c0=0 Length=1
new line.linha11 phases=3 bus1=B10 bus2=B11 r1=1.9030 x1=0.2160 r1=1.9030 x1=0.2160
c1=0 c0=0 Length=1
//definir o SFCR
// pv definition
New PVSystem.PV phases=3 bus1=B10 kV=12.47 kVA=1200 irrad=1.0 Pmpp=1000 model=1
~ temperature=25 effcurve=Eff P-TCurve=PvsT
~ Daily=Irradiancia
//CONTROLE VOLTVAR:
New XYCurve.myvv_curve npts=4 Yarray=(4.0, 0, 0, -0.8) XArray=(0.86, 0.98, 1.01, 1.03)
New InvControl.InvTestPVCtrl mode=VOLTVAR voltage_curve_ref=rated
vvc_curve1=myvv_curve
set maxcontroliter=300
//definir cargas
new load.carga11 phases=1 bus1=B1.1 kv=7.2 kW=161.680 kvar=58.680 Daily=carga
new load.carga12 phases=1 bus1=B1.2 kv=7.2 kW=80.000 kvar=60.000 Daily=carga
new load.carga13 phases=1 bus1=B1.3 kv=7.2 kW=260.000 kvar=147.18 Daily=carga
new load.carga21 phases=3 bus1=B2.1 kv=7.2 kW=265.000 kvar=136.58 Daily=carga
new load.carga22 phases=3 bus1=B2.2 kv=7.2 kW=217.500 kvar=120.97 Daily=carga
new load.carga23 phases=3 bus1=B2,3 kv=7.2 kW=170.000 kvar=105.36 Daily=carga
new load.carga31 phases=3 bus1=B3.1 kv=7.2 kW=64.000 kvar=48.000 Daily=carga
new load.carga32 phases=3 bus1=B3.2 kv=7.2 kW=244.000 kvar=135.180 Daily=carga
new load.carga33 phases=3 bus1=B3.3 kv=7.2 kW=109.000 kvar=69.790 Daily=carga
new load.carga41 phases=1 bus1=B4.1 kv=7.2 kW=180.000 kvar=17.436 Daily=carga
new load.carga42 phases=1 bus1=B4.2 kv=7.2 kW=90.000 kvar=43.590 Daily=carga
new load.carga43 phases=1 bus1=B4.3 kv=7.2 kW=90.000 kvar=43.590 Daily=carga
new load.carga51 phases=1 bus1=B5.1 kv=7.2 kW=232.500 kvar=64.080 Daily=carga
new load.carga52 phases=1 bus1=B5.2 kv=7.2 kW=331.680 kvar=136.140 Daily=carga
new load.carga53 phases=1 bus1=B5.3 kv=7.2 kW=42.500 kvar=26.340 Daily=carga
new load.carga61 phases=1 bus1=B6.1 kv=7.2 kW=47.500 kvar=15.610 Daily=carga
new load.carga62 phases=1 bus1=B6.2 kv=7.2 kW=95.000 kvar=31.220 Daily=carga
new load.carga63 phases=1 bus1=B6.3 kv=7.2 kW=161.680 kvar=58.680 Daily=carga
new load.carga71 phases=1 bus1=B7.1 kv=7.2 kW=95.000 kvar=31.220 Daily=carga
new load.carga72 phases=1 bus1=B7.2 kv=7.2 kW=190.000 kvar=62.450 Daily=carga
new load.carga73 phases=1 bus1=B7.3 kv=7.2 kW=95.000 kvar=31.220 Daily=carga
new load.carga81 phases=1 bus1=B8.1 kv=7.2 kW=90.000 kvar=43.590 Daily=carga
new load.carga82 phases=1 bus1=B8.2 kv=7.2 kW=135.000 kvar=65.380 Daily=carga

```

Quadro 1 – Representação do Sistema Teste da Figura 31 via scripts no OpenDSS.

```

new load.carga83 phases=1 bus1=B8.3 kv=7.2 kW=180.000 kvar=87.180 Daily=carga
// cargas originais
////////////////////////////////////
new load.carga91 phases=1 bus1=B9.1 kv=7.2 kW=95.000 kvar=31.220 Daily=carga
new load.carga92 phases=1 bus1=B9.2 kv=7.2 kW=142.500 kvar=46.840 Daily=carga
new load.carga93 phases=1 bus1=B9.3 kv=7.2 kW=95.000 kvar=31.220 Daily=carga

new load.carga101 phases=1 bus1=B10.1 kv=7.2 kW=135.000 kvar=65.380 Daily=carga
new load.carga102 phases=1 bus1=B10.2 kv=7.2 kW=90.000 kvar=43.590 Daily=carga
new load.carga103 phases=1 bus1=B10.3 kv=7.2 kW=225.000 kvar=108.97 Daily=carga

new load.carga111 phases=1 bus1=B11.1 kv=7.2 kW=175.000 kvar=94.630 Daily=carga
new load.carga112 phases=1 bus1=B11.2 kv=7.2 kW=175.000 kvar=94.630 Daily=carga
new load.carga113 phases=1 bus1=B11.3 kv=7.2 kW=127.500 kvar=79.020 Daily=carga
////////////////////////////////////
Set voltagebases=[12.47]

calcv
solve mode=Daily
/////set mode=Daily number=24 ! --- stepsize=15m ou usar 1min

new monitor.tensao101 element=load.carga101 terminal=1 mode=32
new monitor.tensao102 element=load.carga102 terminal=1 mode=32
new monitor.tensao103 element=load.carga103 terminal=1 mode=32

//new monitor.potenciaPV element=PVSystem.PV mode=1 PPolar=NO terminal=1

new EnergyMeter.Em1 line.linha1
set voltagebases=[12.47]
solve mode=Daily
/////set mode=yearly number=24 ! --- stepsize=15m
//show mon potenciaPV
export monitors tensao101
Plot monitor object=tensao101 channels=(1) bases=[7200]
export monitors tensao102
Plot monitor object=tensao102 channels=(1) bases=[7200]
export monitors tensao103
Plot monitor object=tensao103 channels=(1) bases=[7200]

```

Quadro 1 – Representação do Sistema Teste da Figura 31 via scripts no OpenDSS.

```
//export monitors potenciaPV
//Plot monitor object=potenciaPV channels=(1 3 5)
//Plot monitor object=potenciaPV channels=(2 4 6)
//Plot monitor object=potenciaPV channels=(1)
//////////
//export monitors object=potenciaPV channels=(1)

//////////
```

APÊNDICE B: ATIVIDADES PEDAGÓGICAS USANDO DIAGRAMABARRA

ATIVIDADE 1: APRENDER CONCEITOS DE LINHA DE TRANSMISSÃO, CARGA E GERAÇÃO

Ficha Atividade:	Introdução aos conceitos de sistemas de energia
Objetivo geral:	Conhecer propriedades e características de linhas de transmissão de energia, de geração e de cargas.
Objetivos específicos:	Identificar fatores representativos dos elementos presentes nos sistemas de energia
Objetivos secundários:	<ul style="list-style-type: none"> - Conhecer elementos principais no software de simulação: VSource, BusBar, Line e Load. - Conhecer os parâmetros básicos para simular e resolver problemas de redes de transmissão.
Temas transversais:	<ul style="list-style-type: none"> - Incentivar a criatividade e troca de informação entre parceiros - Despertar o interesse por conhecer realidades do sistema elétrico.
Aprendizados esperados:	<ul style="list-style-type: none"> - Adquirir experiência e a visualizar o aprendido na teoria. - Domínio básico dos componentes básicos do software.
Evidencia de aprendizado:	- Classificar qualitativamente cargas, barra de alimentação e linhas de transmissão.

Esta atividade pretende servir de introdução aos conceitos de sistemas de energia elétrica e ao uso do programa DiagramaBarra. Usa o arquivo de exemplo *Ex01Simple.dsg*, que está incluído nos arquivos instaladores e no site SourceForge. O tutor indicará aos alunos que rodem o programa DiagramaBarra, abram o arquivo citado e explorem o programa. Os estudantes devem descobrir a função das ferramentas, realizar perguntas acerca dos controles, discutir entre parceiros. Posteriormente o professor orientará o curso para o procedimento de simulação e edição de parâmetros. Finaliza a atividade com uma tarefa para o aluno escolher um elemento e realizar variações dos parâmetros a seu critério, tabular os resultados e a partir dos dados, realizar uma inferência do observado.

ATIVIDADE 2: IMPORTÂNCIA DE FATOR DE POTENCIA E CORREÇÃO

Ficha Atividade:	Fator de potência na carga e na linha de transmissão
Objetivo geral:	Conhecer propriedades e características de linhas de transmissão de energia elétrica.
Objetivos específicos:	Identificar fatores na linha de transmissão e na carga que influenciam as perdas na linha.
Objetivos secundários:	<ul style="list-style-type: none"> - Conhecer parâmetros R1, L1, C1 do elemento Line - Conhecer parâmetros KW e KVAR do elemento Load. - Conhecer procedimento para visualizar perdas e potências
Temas transversais:	<ul style="list-style-type: none"> - Incentivar a criatividade e troca de informação entre parceiros - Despertar o interesse por conhecer realidades do sistema elétrico.
Aprendizados esperados:	<ul style="list-style-type: none"> - Adquirir critérios para escolher condutores - Domínio dos componentes básicos do software
Evidencia de aprendizado:	- Classificar qualitativamente condutores de linhas de transmissão com base nas especificações técnicas

Nesta atividade usa-se o arquivo de exemplo *Ex01Simple.dsg*. Como o aluno já explorou a interface, tutor indicará aos alunos uma série de valores a modificar no elemento Line. Os resultados de V_{pu} serão tabulados e os alunos devem discutir o resultado observado. O tutor indicará procedimento para mostrar resultados de potências *Powers* e *Losses*. Pedirá que seja completada a tabela de valores e discutido o observado. Finaliza a atividade com uma tarefa para o aluno procurar um modelo real de linha de transmissão com suas características e custos.

ATIVIDADE 3: CARGA BALANCEADA E DESBALANCEADA

Ficha Atividade:	Carga balanceada e desbalanceada
Objetivo geral:	Conhecer esquemas de conexão trifásica e suas características.
Objetivos específicos:	Identificar problemas de cargas desbalanceadas.
Objetivos secundários:	<ul style="list-style-type: none">- Introduzir ferramenta de dividir fases- Conhecer diferentes formas de conexão monofásica e trifásica
Temas transversais:	<ul style="list-style-type: none">- Motivar o debate de entre parceiros- Conduzir o aprendiz a problemas existentes no sistema elétrico.- Introduzir a legislação vigente referentes a cargas desbalanceadas
Aprendizados esperados:	- Ampliação do domínio dos componentes o <i>software</i>
Evidencia de aprendizado:	- Entrega de tarefas propondo mudanças na interligações entre fases.

Nesta atividade usa-se o arquivo de exemplo incluído *Ex02Unbalanced.dsg*.

Após exposição teórica acerca de fases e desbalanceamento de cargas, o professor indica o uso do arquivo citado, e pede para simular o caso. O aluno já tem domínio dos parâmetros básicos de carga, fontes e linha. O professor propõe uma revisão da normativa da ANEEL e verificam-se os pontos que o sistema cumpre. Como tarefa, professor pede propostas com fundamento de modificações do sistema de energia para, indicando os graus de liberdade nas quais existe factibilidade.

ATIVIDADE 4: FLUXO DE POTENCIA DE UM SISTEMA DE 3 BARRAS

Ficha Atividade:	Fluxo de potencia de um sistema de 3 barras
Objetivo geral:	Aplicar exemplo da literatura
Objetivos específicos:	Conhecer método para resolver problemas com barra PV, PQ e barra de referência.
Objetivos secundários:	<ul style="list-style-type: none">- Introduzir o elemento <i>Generator</i>- Conhecer diferentes formas de conexão monofásica e trifásica- Comprovar níveis de tensão na criação de diagramas
Temas transversais:	<ul style="list-style-type: none">- Motivar liderança- Conduzir o aprendiz a problemas existentes no sistema elétrico.- Introduzir modelos de geração
Aprendizados esperados:	<ul style="list-style-type: none">- Adquirir conhecimento das diferentes formas de geração da matriz energética do SIN (Sistema Interconectado Nacional).- Ampliar domínio dos componentes do <i>software</i>
Evidencia de aprendizado:	- Simulação bem sucedida da rede.

Nesta atividade usa-se o arquivo de exemplo incluído *Ex03Generator.dsg*. Posterior ao professor explicar a matéria de fluxo de potência nas diferentes modalidades de barra (PV, PQ e de referência). Abre-se o arquivo e executa-se a simulação. Este corresponde com o exemplo 7.7 de (ELGERD, 1976). Como tarefa designa-se um aluno como Operador Nacional que distribuirá esquemas de circuitos de 2 ou 3 centrais do SIN de energia convencional motriz, mediante sorteio ou partição por região. Esses conjuntos serão repartidos aos parceiros do curso, para uma simulação distribuída. Posteriormente os dados recopilados dos diferentes subsistemas serão agrupados.

APÊNDICE C: FORMATO ARQUIVOS DE IDIOMAS

Para ampliar o uso do programa DiagramaBarra, o usuário pode modificar o idioma da interface. Os textos de internacionalização estão armazenados na pasta */Lang* e contém arquivos de idiomas no formato XML.

Para adicionar um novo idioma, deve copiar algum dos arquivos, mudar o nome do arquivo e editá-lo com qualquer editor de texto plano com suporte de codificação UTF-8. Cabe a restrição de que não pode conter o mesmo valor de *name* dado que os controles do programa usam esse campo para mostrar no menu. As traduções se realizam no campo *value* de cada elemento *Text*. O Quadro 2 mostra um fragmento de um arquivo de idiomas como exemplo.

Quadro 2 – Fragmento do arquivo de idiomas *Lang/Spanish.xml*

```
<?xml version="1.0" encoding="utf-8" ?>
<DiagramaBarra>
  <Native-Langue name="Spanish" filename="spanish.xml" version="1.0.0">
    <!-- Main Menu -->
    <Text name="fileToolStripMenuItem" value="Archivo"/>
    <Text name="editToolStripMenuItem" value="Editar"/>
    <Text name="simulationToolStripMenuItem" value="Simulación"/>
    <Text name="toolsToolStripMenuItem" value="Herramientas"/>
    <Text name="helpToolStripMenuItem" value="Ayuda"/>
    <!-- File Menu -->
    <Text name="newToolStripMenuItem" value="Nuevo"/>
    <Text name="openToolStripMenuItem" value="Abrir archivo"/>
    <Text name="importFromDssToolStripMenuItem" value="Importar de archivo DSS"/>
    <Text name="saveToolStripMenuItem" value="Guardar archivo"/>
    <Text name="saveAsToolStripMenuItem" value="Guardar archivo como.."/>
    <Text name="exitToolStripMenuItem" value="Salir"/>
    <!-- Edit Menu -->
    <Text name="connectToolStripMenuItem
```

Fonte: Produção do próprio autor